Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería en Informática

Proyecto Fin de Carrera

Plataforma de Auditoría Continua de Sistemas de Información basada en el análisis de riesgos y Business Intelligence

Autor: Luis Gabriel Tablado Félix

Tutor: Juan Miguel Gómez Berbís

Cotutor: Ángel Lagares Lemos

Leganés, Marzo de 2015

'Why do you ask so many questions, Mau?'

'Because I want so many answers!'

- Sir Terry Prattchet, *Nation*

Agradecimientos

Este texto de agradecimientos debería remontarse, como poco, a 2001 cuando empecé a coger el tren de cercanías para ir a la universidad, pero desde entonces ha pasado tanto tiempo que la lista acabaría por tener poco sentido.

En su famoso discurso de Stanford, Steve Jobs hablaba sobre cómo, si miras hacia atrás y conectas los puntos que hubo en tu camino pequeñas decisiones conectarán en tu futuro gracias al karma, al destino o a lo que sea. Bien, esa es una historia genial para contar cuando estás subido delante de un atril y has triunfado en la vida pero es una gran mentira. No creo que retrasar este proyecto haya sido una buena decisión o que el destino me haya llevado hasta aquí. No creo que pueda unir una serie de puntos plateados que ahora me hagan pensar que este proyecto es mejor de lo que hubiera sido. Sin embargo, hoy estoy extremadamente contento y orgulloso de haber terminado el camino que empecé en 2001 y en los momentos de felicidad hay que dar las gracias y aunque no de nombres. Todos sabéis de sobra quienes sois, si es que un día leéis esto.

Primero, a mi familia, que, desde que era pequeño, incluso en los baches han empujado para que tenga una formación, una cultura, que es tu único patrimonio real. Después a los compañeros de la universidad por ayudar a que esos 5 años en Leganés fueran un continuo bombardeo de inquietudes. A mis tutores, por aceptar locas propuestas con prisas que nos libren a todos de la extinción. A los que han estado caminando, corriendo y peleando conmigo o contra mí estos años, gracias por estar ahí. Y a los que no están, que paradójicamente, serían los más felices si tuvieran este texto en las manos.

Isaac Newton decía que él vio más por estar subido a hombros de gigantes, bueno, yo estoy aquí hoy por estar en vuestros hombros muchas veces y dado que ninguno sois gigantes, imagino que tiene más mérito.

Y por último, gracias a ella, que lo cambió todo.

Resumen

La Auditoría de Sistemas de la Información es un campo de la Informática que no suele ser tenido en cuenta en entornos académicos. Su rol de tarea de cumplimiento normativo tradicionalmente la ha relegado a un segundo plano dentro de las disciplinas que se tratan en Ingeniería Informática.

Más allá del puro rol de cumplimiento la Auditoría de Sistemas de la Información proporciona a las organizaciones un mayor control sobre sus procesos de tecnología de la información así como una oportunidad para la mejora de su operativa habitual.

Tradicionalmente la auditoría se ha basado en tareas discretas en las que se revisaban de forma periódica diversos campos de la tecnología de las organizaciones. En los últimos años se está imponiendo una tendencia en la que se intenta proponer métodos automáticos de obtención y análisis de la información que permitan que el trabajo del auditor sea un proceso continuo haciendo que los resultados de las pruebas de auditoría y los datos que las soportan estén disponibles prácticamente en tiempo real. Este proceso se denomina Auditoría Continua.

En el presente Proyecto de Fin de Carrera se ha diseñado e implementado una aplicación online basada en la solución *Open Source* de *Business Intelligence* SpagoBI que proporcione servicios de auditoría continua para una organización.

La aplicación *auditop continuous audit* es una solución web que permite la carga, transformación y validación de datos heterogéneos manual o automáticamente mediante su integración con la solución ETL Talend Open Studio, la definición de pruebas de auditoría y el cálculo periódico de éstas y la definición y generación de reportes basados en la tecnología BIRT.

En este Proyecto de Fin de Carrera se ha realizado un análisis de los riesgos que podría tener una organización tipo y se ha propuesto un conjunto de pruebas de auditoría y reportes asociados para demostrar el funcionamiento de la aplicación en un entorno corporativo real.

Abstract

Information Systems Audit is a field in Computer Science that is not usually of a great importance in academic environments. The usual role of Information System Audit has been compliance and has traditionally relegated it into the background between the topics in Computer Science Engineering.

Beyond this compliance role Information Systems Audit gives organizations a greater control over their technological processes as well as an opportunity for improvement to their business as usual.

Traditionally audit has been based on discreet tasks where several fields of the technology in an organization were reviewed periodically. Lately a new trend is growing and automatic methods to obtain and analyze the information are being implemented. With these new methods the auditor work is becoming a continuous process and the results of audit tests and the data they are based on are available in near real time. This process is called Continuous Audit.

In this Final Degree Project an application based on the Open Source Business Intelligence platform SpagoBI has been designed and implemented. This solution can provide continuous audit services for a given organization.

The application *auditop continuous audit* is a web solution that allows heterogeneous data load, transformation and validation through the integration with the ETL solution *Talend Open Studio,* furthermore audit tests can be defined and periodically calculated and reports based on BIRT technology can be defined and generated.

In this Final Degree Project we have carried out a risk assessments on an hypothetical organization and we have proposed a set of audit tests with associated reports to demonstrate the functionality of the application in a real corporate environment.

Esta página se ha dejado intencionadamente en blanco

Índice General

Índice de ilustraciones

Índice de tablas

Esta página se ha dejado intencionadamente en blanco

Capítulo 1: Introducción

En este primer apartado se procede a enunciar las razones que han motivado el desarrollo del presente proyecto, haciendo especial hincapié en la relevancia y notable actualidad de los conceptos implicados. A continuación, se presenta la estructura de la memoria y se detallan los aspectos tratados en cada uno de los apartados que la conforman.

Como último punto de la introducción son enumeradas y descritas las definiciones de los términos, las abreviaturas y los acrónimos empleados a lo largo de todo el documento.

1.1 Descripción del problema

La Auditoría de Sistemas es un procedimiento mediante el cual se analiza y valora el conjunto de sistemas de la información que prestan servicio a una organización. La Auditoría de Sistemas se ha convertido en una faceta ineludible en la gestión de los sistemas de información de una organización, sobre todo, en entornos altamente regulados.

La Auditoría de Sistemas es un campo relativamente joven. Sin embargo no está exento de evolución y de influencias, sobre todo en los últimos años con los cambios normativos y la emergencia de reguladores transnacionales especialmente en el sector bancario. Esto ha provocado que el enfoque tradicional de auditorías periódicas quede un poco desfasado con respecto a las necesidades de una organización compleja.

Tradicionalmente se ha abordado la auditoría como un proceso discreto, revisándose de forma periódica un campo concreto de la organización, emitiendo un informe de auditoría que estará en vigor hasta la siguiente revisión de auditoría independientemente de los cambios que hayan acaecido en la organización Esto hace que incidencias relevantes puedan ser pasadas por alto o que se planifiquen futuras auditorías basándose en información no actualizada.

Para poder adaptarse a las exigencias regulatorias y a entornos tecnológicos cambiantes la auditoría de sistemas debe variar. Es más, la cantidad de información disponible de forma automática ha aumentado exponencialmente con la implantación de sistemas de *Business Intelligence, Data Warehousing y Big Data.* Ahora se dispone de grandes conjuntos de información y de capacidad de análisis si se diseñaran los sistemas automáticos que se hicieran cargo de este análisis. Actualmente pocos departamentos de

auditoría disponen de estos medios automáticos teniendo que recurrir a muestreos sobre la población de los datos para poder aprovechar esta información. Sobra comentar que este procedimiento pierde información relevante por el camino aparte de introducir riesgo por la manualidad del proceso.

1.2 Motivación

Ante las dos vertientes del problema comentadas en el apartado anterior este Proyecto de Fin de Carrera pretende proponer una solución mediante un sistema automático integrado que adquiera y transforme los datos procesables relevantes de la variedad de sistemas de información de la empresa y realice pruebas de auditoría continua sobre ellos proporcionando información fiel, actualizada y relevante sobre el estado de los sistemas de información de una organización.

Como se verá más adelante este tipo de sistemas son uno de los puntos hacia donde más inversión e investigación se está realizando dentro del campo de la auditoría y no solamente en la auditoría de sistemas de información. Es por ello que nos pareció muy interesante personal y profesionalmente realizar un análisis pormenorizado de la investigación que se está realizando en este campo y diseñar e implementar un sistema de auditoría continua.

Uno de los campos de mayor interés personal del autor de este Proyecto de Fin de Carrera es el Software Libre. Lamentablemente el uso de software libre en el campo de la auditoría de sistemas de la información es, por decirlo de forma misericordiosa, escaso. En la experiencia profesional del autor el uso de Software Libre por parte de las corporaciones relacionadas con la auditoría de sistemas de información tiene un enfoque parasitario. Las corporaciones se aprovechan del Software Libre pero se les olvida contribuir a la comunidad con los beneficios que han obtenido.

Por ello uno de los condicionantes de diseño de la aplicación innegociable será el uso de componentes de Software Libre.

1.3 Objetivos del trabajo

Este Proyecto de Fin de Carrera plantea tres objetivos fundamentales que pretendemos cubrir detalladamente en esta memoria.

- **Analizar la auditoría de sistemas –** Proporcionando un estudio histórico de la función de la auditoría de sistemas y un detalle de los procedimientos y técnicas de auditoría de sistemas de información. El objetivo será que al finalizar la lectura del documento el lector haya adquirido un conocimiento general sobre la auditoría de sistemas y, con suerte, motivar al lector a que continúe profundizando en el aprendizaje por su cuenta y reoriente su carrera profesional hacia este apasionante campo

- **Analizar la auditoría continua –** Identificando las necesidades que llevan a implementar procesos de auditoría continua, comentando las líneas de investigación que hay sobre el tema y conociendo los pasos que hay que seguir para diseñar e implementar un sistema de auditoría continua.

- **Diseñar e implementar una solución de auditoría continua -** El objetivo final de este Proyecto de Fin de Carrera es presentar una solución 100% funcional basada en componentes de Software Libre que se pueda utilizar en entornos de producción para realizar tareas de auditoría continua.

1.4 Estructura del documento

En este primer capítulo se establece una introducción al presente trabajo, así como una recopilación de definiciones, abreviaturas y acrónimos útiles para el entendimiento de este documento.

En el segundo capítulo se presenta una descripción de la auditoría de sistemas, incluyendo una reseña histórica, información sobre los ámbitos que cubre la auditoría de sistemas y los tipos de pruebas que se realizan.

En el tercer capítulo se analiza la auditoría continua, revisando las líneas de investigación actuales sobre auditoría continua, la dirección de la industria y las técnicas que se aplican para realizar sistemas de auditoría continua.

En el capítulo cuarto se realiza una descripción en profundidad del sistema implantado desde el análisis del sistema (Casos de uso y requisitos) hasta el diseño de la solución implementada. También se incluye un análisis sobre las tecnologías valoradas para la implementación del sistema.

En el capítulo cinco se presentan unos ejemplos de uso de la principal funcionalidad del sistema

El capítulo sexto contiene información sobre la gestión del proyecto, control de las tareas y tiempos del mismo y presupuesto.

El séptimo capítulo presenta las conclusiones a las que hemos llegado con la realización del proyecto y las líneas de trabajo futuro que planteamos para continuar desarrollando la solución propuesta.

Los capítulos ocho nueve presentan una lista de referencias utilizadas para la realización de este proyecto así como una serie de anexos a la memoria de este proyecto de fin de carrera.

1.5 Definiciones, Abreviaturas y Acrónimos

En los siguientes subapartados son enumeradas y descritas las definiciones de los términos, las abreviaturas y los acrónimos empleados durante de toda la memoria.

1.5.1 Definiciones

Base de Datos	Una base de datos es una colección de datos organizados. Habitualmente una base de datos está asociada a un sistema gestor de base de datos que permite consultar la base de datos y gestionar el almacenamiento de la información.
Datacenter	Un datacenter o centro de proceso de datos es una instalación en la que se encuentran los activos de información (Servidores, almacenamiento de datos y equipamiento de red) de una organización
Open Source	Open Source o Código abierto es la expresión con la que se conoce al software distribuido y desarrollado bajo una licencia libre que permite el acceso, modificación y distribución del código fuente.
Outsourcing	Es el proceso mediante el cual una empresa externaliza una parte de su actividad, es decir, contrata a una empresa externa para gestionar un proceso habitualmente no principal.

Risk Assessment	Es la determinación del valor cuantitativo o cualitativo del riesgo asociado a una situación y amenaza concretas
Walkthrough	Recorrido paso a paso por un proceso para, utilizando el conocimiento de la persona encargada del proceso, poder entender el proceso y detectar debilidades o errores.

Tabla 1 – Definiciones

1.5.2 Acrónimos

ACPA	Instituto Americano de Contables Públicos Certificados
AMA	Advanced Measurement Approach
BI	Business Intelligence
CAAT	Técnicas de auditoría asistidas por computadora
CISA	Certified Information Systems Auditor
CobiT	Control Objectives for Information and Related Technology
COSO	Committee of Sponsoring Organizations of the Treadway Commission
CSV	Comma Separated Values
EAM	Módulo de auditoría Integrado
EBA	Regulador bancario único europeo (European Banking Authority)
ACID	Atomicity, Consistency, Isolation, Durability
AGPL	Affero General Public License
BIRT	Business Intelligence and Reporting Tools
EDP	Proceso Electrónico de Datos (Electronic Data Processing)
EDPAA	Electronic Data Processing Auditors Association
ERP	Enterprise Resource Planner
ETL	Extract Transform Load
GAS	Software generalizado de auditoría (Generalized Audit Software)
GNU	GNU is not Unix
GPL	GNU General Public License
HTTP	Hypertext Transport Protocol

ISACA	Information System Audit and Control Association
ITAF	IT Assurance Framework
JVM	Java Virtual Machine
KPI	Key Performance Indicator
KRI	Key Risk Indicator
LGPL	GNU Lesser General Public License
LOPD	Ley Orgánica de Protección de datos
MPL	Mozilla Public License
OTS	Request for changes
PDF	Portable Document Format
RDA	Risk Data Agreggation
RFC	Request for Changes
SEC	US Security and Exchange Comission
SI	Sistemas de Información
SOX	Ley Sarbanes-Oxley (“Public Company Accounting Reform and Investor Protection Act”)
SQL	Structured Query Language
TOS	Talend Open Studio
URL	Uniform Resource Locator

Tabla 2 - Acrónimos

Capítulo 2: La auditoría de Sistemas de Información

En este apartado se realizará una descripción general de la auditoría de Sistemas de la Información y del proceso de realización de una auditoría.

2.1 Introducción

Como se indica en [1] la auditoría de sistemas de información [Auditoría de SI de ahora en adelante] se define como los procedimientos y técnicas que se usan para la evaluación y control de un sistema de información, con el objetivo de validar si sus actividades son las adecuadas en relación con la normativa vigente y los objetivos que haya fijado la organización. Una auditoría de sistemas debe obtener, agrupar y evaluar una serie de evidencias para determinar si el sistema salvaguarda los activos, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficazmente los recursos,

La auditoría informática también se encarga del aseguramiento continuo de que los recursos informáticos operan en un ambiente de seguridad y control eficientes, con la finalidad de proporcionar a la alta dirección la certeza de que la información utilizada por el negocio cumple los criterios básicos integridad, calidad, exactitud, confiabilidad, etc.

El proceso de la auditoría tiene como propósito principal evaluar los recursos (humanos, materiales, financieros, tecnológicos, etc.) relacionados con la función de sistemas de información para garantizar al negocio que toda la operativa funcionará con unos niveles de integración y rendimiento satisfactorios para cumplir los objetivos de productividad y rentabilidad de la empresa

2.1.1 Historia de la auditoría de Sistemas de Información

El nacimiento de la Auditoría de Sistemas está íntimamente ligado a la historia de la contabilidad. La Auditoría de Sistemas de la Información tiene sus orígenes en la Auditoría del Proceso Electrónico de Datos (EDP) y surgió como resultado del creciente uso de la tecnología en sistemas contables. La necesidad de control sobre las Tecnologías de la Información y el creciente foco en la ciberseguridad han incrementado las funciones de la auditoría de SI. En los últimos años con el aumento de regulación a resultado de la crisis financiera de 2008 el rol del auditor de SI es cada vez más importante tanto como auditor

de SI independiente como en colaboraciones con la auditoría clásica debido al conocimiento transversal de los sistemas que puede aportar.

En los años 50 comenzó en Estados Unidos la introducción de sistemas de procesamiento de datos. Desde el punto de vista de la auditoría, los primeros sistemas relevantes son los sistemas computarizados de contabilidad que fueron usados por primera vez en *General Electric* en 1954. Sin embargo en esta primera década, hasta mediados de los años 60, la auditoría gravitaba alrededor del ordenador no hacía uso real del mismo, ni mucho menos se realizaba un análisis detallado. En esta época la tecnología estaba dominada por los grandes mainframes y poca gente tenía las habilidades para programarlos.

Esto empezó a cambiar a finales de los años 60 con la introducción de máquinas más pequeñas y económicas. Esto incrementó del uso de ordenadores en los negocios y el uso de sistemas de EDP se hizo común en todos los negocios. A partir de este momento el auditor, que hasta este momento tenía una formación contable tuvo que adquirir conocimientos sobre el procesado electrónico de datos. Con el uso extendido de los ordenadores el ecosistema de sistemas de contabilidad creció exponencialmente. Para adaptarse a ello la industria de la auditoría tuvo que desarrollar su propio software y con ello nacieron los primeros sistemas generalizados de auditoría (GAS).

En 1968 el Instituto Americano de Contables Públicos Certificados (AICPA) juntó a las principales firmas de contabilidad del momento (En aquella época eran conocidas como las *Big Eight* hoy en día tras un sinfín de fusiones y cambios de nombre son conocidas como las *Big Four*) para desarrollar un marco de control para la auditoría de los sistemas EDP. El resultado fue la publicación de ”*Auditing & EDP”.* Este libro era una guía que trataba acerca de cómo documentar auditorías de sistemas EDP y ejemplos sobre cómo procesar las revisiones de control interno.

Sobre esta época los auditores de EDP fundaron la *Electronic Data Processing Auditors Association* (EDPAA). Esta asociación tenía como objetivo la producción de guías, procedimientos y estándares para la realización de auditorías de EDP. En 1977 se publicó la primera edición de *Control Objectives*. Esta publicación es muy conocida a fecha de hoy bajo el nombre *Control Objectives for Information and Related Technology (CobiT).* Cobit es un conjunto de objetivos de controles IT generalmente acepados en la industria para los auditores de IS. En 1994 EDPAA cambió su nombre a *Information Systems Audit and*

Control Association (ISACA), que a fecha de esta publicación sigue siendo la principal asociación de auditores de sistemas y proporciona un conjunto de estándares y certificaciones sobre las auditorías de IS.

Desde los años 60 hasta la actualidad la auditoría de IS ha tomado entidad propia y se ha independizado de sus orígenes accesorios a la contabilidad. En la actualidad, con el crecimiento de Internet y el comercio electrónico la auditoría de IS tiene muchas más atribuciones y funciones ya que las tecnologías de la información son un pilar básico en el funcionamiento de todas las organizaciones. Centrándonos en Internet a auditoría de IS ayuda a las organizaciones con presencia en Internet a asegurar sus procesos sin lastrar el crecimiento del comercio y las comunicaciones.

Como comentábamos anteriormente en el comienzo del siglo XXI han ocurrido eventos que han reforzado el papel de la auditoría de IS.

Por un lado, en 2001 el escándalo contable de Enron hizo patente las limitaciones del modelo de auditoría imperante en ese momento y los conflictos de interés que había en el doble rol de Auditor/Consultor de las *Big Four*. La solución propuesta por el gobierno americano ha supuesto un incremento en las tareas de control interno de las organizaciones y un reporting continuo a los reguladores como la Sociedad del mercado de valores americana (SEC), así, con la ley *Sarbannes-Oxley* (SOX) los auditores de IS tienen como obligación la revisión de todos los objetivos de control interno y la elaboración de informes periódicos.

Por otro lado, tras la crisis financiera de 2008 ha supuesto una ola de nuevas normativas de control especialmente para las entidades financieras. Dentro de las nuevas normativas de control del riesgo operacional, del riesgo de crédito u del capital financiero (AMA [2], RDA [3]y Basilea II y III[1]) se están estableciendo sistemas informáticos cada vez más complejos que aglutinen distintos tipos de información operacional y financiera que deben ser revisados de forma periódica tanto por el interés de la organización como por interés de los reguladores..

Para terminar, en el año de esta publicación (2015) se ha producido un cambio que va a afectar la forma de realizar auditorías de SI en entidades financieras en España. Dentro de las medidas tomadas para la armonización de la eurozona la supervisión de las

[1] http://www.bis.org/bcbs/basel3.htm

entidades financieras ha sido transferida por el Banco de España al regulador único bancario europeo (EBA). Este movimiento traerá, sin duda nuevas condiciones y regulaciones a las que los auditores tendrán que adaptarse.

2.1.2 Principales estándares y organismos certificadores de auditoría de Sistemas de Información

En este apartado se describirán los estándares y organismos internacionales más importantes en el ámbito de la auditoría de SI

2.1.2.1 ISACA

Como se comentaba en el apartado acerca de la historia de la auditoría, ISACA es la principal asociación de auditores de IS. Con un bagaje que se remonta a 1969 cuando se estableció la Asociación de Auditores de EDP (EDPAA), toma el nombre de ISACA en 1994.

Esta asociación proporciona una serie de guías, políticas, procedimientos y estándares sobre la auditoría interna y valida una serie de certificaciones profesionales en el ámbito de la gestión de auditoría, riesgos y seguridad de sistemas de información. La principal certificación aceptada en la industria para un auditor de SI es CISA (Certified Information Systems Auditor). Actualmente 88.000 auditores en todo el mundo están certificados por ISACA como CISA.

2.1.2.2 ITAF

ISACA ha desarrollado un marco de control de auditoría que se conoce por el acrónimo ITAF. Information Technology Assurance Framework (Marco de control sobre el aseguramiento de la tecnología de la información) [4] es un modelo de buenas prácticas que

- Proporciona una guía para el diseño, la realización y el reporte de auditorías IT y trabajos de aseguramiento.
- Define términos y conceptos específicos del aseguramiento de IT.
- Establece estándares que atañen a la auditoría IT, los roles y responsabilidades que desempeñan los profesionales IT y el código de conducta esperable en los profesionales.

Dentro del marco de control ITAF se definen una serie de estándares para cada uno de los campos que cubre que están disponibles en el dominio público.

Los últimos estándares de auditoría de ISACA a fecha de esta publicación son los siguientes.

1001 Estatuto de la función de auditoría
1002 Independencia organizacional
1004 Expectativa razonable
1005 Debido cuidado profesional
1006 Competencia
1007 Afirmaciones
1008 Criterios
1201 Planificación de la asignación
1202 Evaluación de riesgo en planificación
1203 Desempeño y supervisión
1204 Materialidad
1205 Evidencia
1206 Uso del trabajo de otros expertos
1207 Irregularidades y actos ilegales
1401 Reportes
1402 Actividades de seguimiento

Tabla 3 – Estándares de auditoría y aseguramiento SI

Para la implementación de estos estándares ITAF proporciona unas guías que una organización puede utilizar para realizar las labores de auditoría de acuerdo a las mejores prácticas.

2001 Audit Charter
2002 Organisational Independence
2003 Professional Independence
2004 Reasonable Expectation
2005 Due Professional Care
2006 Proficiency
2007 Assertions
2008 Criteria
2201 Engagement Planning
2202 Risk Assessment in Audit Planning
2203 Performance and Supervision
2204 Materiality
2205 Evidence
2206 Using the Work of Other Experts
2207 Irregularity and Illegal Acts
2208 Audit Sampling
2401 Reporting
2402 Follow-up Activities

Tabla 4 – Guías de auditoría y aseguramiento SI

2.1.2.3 COBIT

Uno de los principales estándares acerca de la gestión y el control del *governance de IT de las organizaciones son l*os Objetivos de Control para Información y Tecnologías Relacionadas (COBIT, en inglés: Control Objectives for Information and related Technology).

COBIT se presenta en una guía de mejores prácticas descrita como un marco de referencia, enfocado en el control y supervisión de tecnología de la información. Mantenido por y el IT Governance Institute, tiene una serie de recursos que pueden servir de modelo de referencia para la gestión de TI, incluyendo un resumen ejecutivo, un framework, objetivos de control, mapas de auditoría, herramientas para su implementación y principalmente, una guía de técnicas de gestión.

Aunque el COBIT no se utiliza como guía para la realización de auditorías, es relevante ya que muchas empresas utilizan los principios y controles definidos en el COBIT como referencia para implementar su modelo de control interno.

A fecha de esta publicación la versión más actualizada de COBIT es COBIT 5. COBIT 5 ayuda a que la IT se gobierne y gestione de forma coherente en toda la organización, cubriendo la totalidad de áreas de negocio y áreas funcionales de responsabilidad de IT considerando los intereses de los actores internos y externos de la organización,

COBIT 5 se basa en 5 principios clave para gestionar el *governance* de la IT de la organización.

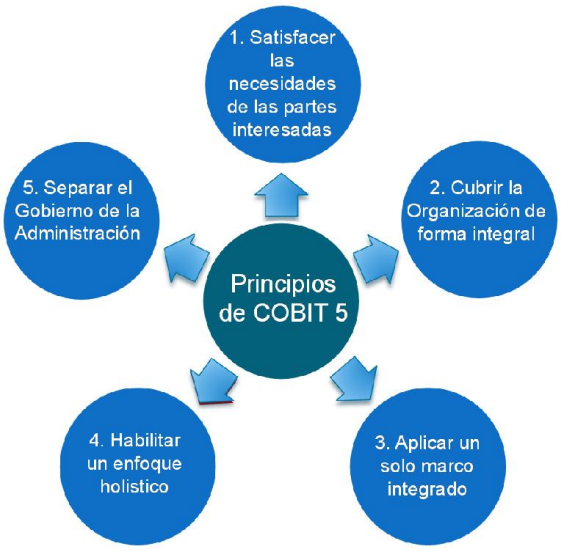


Ilustración 1: Principios de COBIT 5

En el sitio web de ISACA se pueden consultar materiales acerca la implementación práctica de estos 5 principios y su adaptación práctica a una organización. [5]

2.2. Objetivos generales de la auditoría de Sistemas de Información

Tradicionalmente, los objetivos tradicionales de una auditoría de SI son la protección de activos, la integridad de datos y garantizar la eficacia y eficiencia en la gestión de los mismos.

Protección de los datos

Se debe destacar que el control de la protección de los activos en la auditoría de SI debe abarcar tres campos principales:

- **Seguridad física**. Son todos los elementos accesorios al sistema que, aunque no se consideran parte integral del sistema pueden causar problemas en el mismo si están ausentes o fallan. En este campo se encuentran la seguridad e ininterrumpibilidad de los suministros, el control de medio ambiente de los sistemas y la seguridad física de las instalaciones
- **Confidencialidad y seguridad de la información.** Abarca la salvaguarda y el control del acceso a la información.
- **Cumplimiento y aspectos económicos.** La auditoría debe garantizar que la normativa vigente en materia de sistemas de información y protección de los datos se cumple, no sólo realizando las revisiones periódicas estipuladas sino garantizando que se siguen pasos orientados a obtener este cumplimiento en toda la gestión de SI. También es tarea de auditoría valorar que la gestión de los SI se realiza teniendo en cuenta la rentabilidad económica de los activos y de las medidas de protección

Integridad de los datos

La eficacia de un sistema de información se valora midiendo la aportación a la organización de una información válida, exacta, completa, actualizada y disponible en el momento adecuado y para el personal adecuado. La medición debe realizarse en términos de calidad, integridad, plazo y coste.

No se debe pasar por alto que esta medición se debe realizar tanto en la entrada como en la salida de los procesos. Sin validar los datos que entran en el sistema no se puede valorar que los datos ya procesados son correctos ya que no se puede localizar en qué punto del procesamiento suceden los errores

Rentabilidad y cumplimiento

Una auditoría de SI debe revisar que el sistema mantiene un equilibrio entre los riesgos asumidos, costes de medidas mitigadoras y los costes de operación del sistema asegure el equilibrio entre riesgos, costes de seguridad y los costes del propio sistema. Un auditor debe valorar siempre en sus análisis que el coste de las medidas de seguridad o medidas correctivas que recomiende no puede ser mayor al valor de los activos que analiza o mayor a la pérdida esperada en caso que el riesgo se materialice.

La optimización de la rentabilidad de un sistema de información se fundamenta en estos tres pilares

- **Evaluación de los costes reales**. Se deben contabilizar los costes en los que incurre la organización por la gestión de sus sistemas de información, valorando los costes asociados al sistema en sí (hardware, software, mantenimiento) así como los costes inherentes a los procesos en los que los sistemas de información forman parte.
- **Comparación de los costes del sistema con magnitudes económicas de la organización.** Este benchmarking proporcionará información importante a la dirección sobre la rentabilidad real que proporciona a la organización la explotación de los sistemas de información. Esta información es de gran ayuda para justificar las inversiones en sistemas de información que, demasiadas veces, son valorados sólo como un coste y no como una inversión
- **Comparación de los costes con empresas similares.** Esta comparativa servirá para medir la adecuación de la inversión en SI. El auditor deberá manejar información de la industria como informes de consultoría estratégica para poder comparar en qué punto de evolución tecnológica se encuentra la empresa y poder ayudar así en las decisiones estratégicas

2.3. Tipos de auditoría de sistemas de información

La gran mayoría de las auditorías de sistemas de información no se plantean como análisis globales en los que se obtiene una valoración de toda la tecnología de la organización. Para optimizar el coste y poder profundizar en el análisis de los sistemas habitualmente se analiza un dominio concreto de los sistemas de información o una aplicación seleccionada.

Para proporcionar una clasificación de las áreas de la auditoría de sistemas de la información tomamos la segmentación propuesta por ISACA que actúa como la principal autoridad de certificación de auditores de SI.

ISACA [6] propone una clasificación detallada de las auditorías haciendo hincapié en la relación con los procesos de negocio. Los diferentes tipos de auditorías de SI serían:

- **Auditorías de cumplimiento.** Las auditorías de cumplimiento incluyen la validación de los controles que se han establecido en la organización para adherirse a la

regulación vigente o a los estándares de la industria. Ejemplos de este tipo de auditorías son las auditorías de cumplimiento de la Ley Orgánica de Protección de Datos (LOPD) o las de cumplimiento de la ley Sarbanes-Oxley (SOX) para empresas que cotizan en el mercado bursátil americano.

- **Auditorías financieras.** Estas auditorías se suelen realizar como colaboración con un departamento de auditoría contable. Aseguran la validez del reporting financiero de la organización. Un auditor de SI puede proporcionar una validación detallada de los estados financieros de la compañía en lugar del enfoque tradicional de pruebas por muestreo. Un ejemplo de esta auditoría sería una prueba de recalculo de los estados financieros a partir de la información contable de una empresa.
- **Auditorías operacionales.** Evalúa la estructura de control interno de la organización en un proceso determinado o un área. Un ejemplo de estas auditorías sería una auditoría de la gestión de la seguridad lógica de los sistemas.
- **Auditorías de procesos.** Orientadas a detectar problemas en la eficiencia de la productividad operativa dentro de una organización. Estas auditorías están cobrando cada vez más importancia como fuente de información en la optimización de procesos.
- **Auditorías de sistemas de información.** Recogen y evalúan evidencias para determinar si los sistemas de información salvaguardan de forma adecuada los activo, mantienen la integridad y disponibilidad de los datos y los sistemas, proporcionan información relevante y confiable, consiguen los objetivos de la organización de forma efectiva, utilizan los recursos de la organización de forma eficiente y tienen controles internos que proporcionan seguridad razonable de que los objetivos de negocio, operacionales y de control se alcanzan. También cubren las medidas que toma la organización para prevenir, detectar y corregir eventos no deseados que puedan poner en riesgo los sistemas de información.
- **Auditorías especializadas.** Dentro de la categoría de auditorías de SI hay una subcategoría que está tomando cada vez más relevancia y merece la pena destacar. Las auditorías de servicios prestados por terceras personas. Las organizaciones, cada vez más, recurren al *outsourcing* de las actividades relacionadas con la tecnología y es responsabilidad de auditoría validar estas actividades y asegurar la existencia de controles internos que aseguren el cumplimiento de los servicios contratados.

- **Auditorías forenses.** Especializadas en el descubrimiento, la detección y el seguimiento de fraudes y crímenes. Su principal cometido es la obtención de evidencias para su disposición por parte las autoridades legales y judiciales. Por ello este tipo de auditorías deben, ante todo, asegurar la preservación de las evidencias sin ningún tipo de modificación.

Esta división de auditorías de sistemas de información es la más general y cubre todo el espectro del trabajo que un auditor informático realiza dentro de una organización. De todas las categorías listadas anteriormente, el alcance de nuestro proyecto se centra en las auditorías de SI propiamente dichas, así que procederemos a un análisis más detallado de este tipo de auditorías.

2.3.1 Áreas de la Auditoría de los Sistemas de la Información

1. Planificación y Gestión

Se ha de validar la organización en todos sus aspectos: organigrama, distribución de funciones y comunicación con las áreas de la organización. Además, se deberá tener en cuenta la gestión y dirección realizada por el área de IT, evaluando como impactan en la consecución de los objetivos del negocio a través de la visión estratégica, la planificación y la administración de los recursos tecnológicos.

- **Organización y Estrategia**: La estructura organizativa debe ser adecuada; claramente identificada y conocida. Debe garantizar la efectividad de las actividades de Tecnología con el resto de áreas.

  La organización debe disponer de una metodología formal de gestión de IT, que contemple procedimientos adecuados para la elaboración del plan de sistemas, prestando atención a la concordancia entre la estrategia de negocio y los recursos de IT asignados.
- **Control y Recursos:** Deben existir procedimientos para la generación y seguimiento de los presupuestos donde se verifique la alineación de los mismos con el plan de sistemas.

  La relación con los proveedores debe estar formalizada mediante un contrato de prestación de servicios, donde se permitan identificar y medir de forma fidedigna el

grado de cumplimiento de las obligaciones contratadas. Adicionalmente se debe contar con las herramientas adecuadas para su seguimiento.

Los requisitos normativos, tanto externos como internos, deben ser identificados y tenidos en cuenta por el Governance de IT para su posterior análisis de cumplimiento, existiendo mecanismos de control para ello.

2. Producción

La explotación y administración de los sistemas debe proporcionar unos niveles de eficacia, eficiencia y rendimiento adecuados de acuerdo a los requerimientos del negocio, asegurando la continuidad de las operaciones.

- **Gestión y control:** Se debe determinar la estructura organizativa de las áreas productivas, verificando si existe una correcta segregación funcional con el resto de áreas. El servicio de producción debe contar con la identificación de los activos involucrados y de los indicadores de calidad, que permitan una medición del nivel del servicio definido. En este sentido se ha de verificar la evaluación periódica del mismo y de su cumplimiento.
- **Operación**: Se revisará la planificación de tareas y los procesos periódicos existentes, verificando que se realiza de forma eficiente, apoyándose en automatismos que minimicen el riesgo operativo en la gestión de la plataforma.
- **Monitorización:** verificar que se realiza un adecuado seguimiento y monitorización de los sistemas y comunicaciones, prestando atención al funcionamiento, rendimiento y capacidad, y posibilitando la toma eficaz de medidas reactivas y preventivas.
- **Mantenimiento y soporte:** Se debe identificar y evaluar los recursos y procedimientos reservados para la gestión de peticiones, incidencias y cambios sobre la plataforma en producción.
- **Respaldo:** Se comprobará que la realización y mantenimiento de copias de seguridad se lleva a cabo de acuerdo con la política de respaldos definida; verificando el cumplimiento de requisitos generales y específicos de la plataforma y sus aplicativos, evaluando la idoneidad de la configuración de los respaldos y posibles lugares de almacenamiento externos.
- **Contingencias:** Se debe verificar la existencia de un plan de recuperación de sistemas (plan de contingencias), que asegure la continuidad de las operaciones

críticas de negocio en caso de desastre y que esté alineado con el plan de continuidad del negocio.

Adicionalmente, se verificará la realización de pruebas periódicas de contingencia de la plataforma, verificando que se cumple con los tiempos de recuperación establecidos en el Plan de Continuidad de negocio, de la existencia de la documentación de las pruebas y del análisis de sus resultados.

3. Seguridad

El acceso y la manipulación indebida o malintencionada de los activos de información de la organización, pueden provocar graves pérdidas económicas, de imagen y reputación así como de cumplimiento de la normativa vigente. Se debe garantizar la existencia de suficientes medidas de protección, tanto físicas como lógicas que garanticen la seguridad de esta información. Las deficiencias sobre estos aspectos inciden en los siguientes factores:

- **Gestión y Control de la Seguridad**: Las responsabilidades y gestión de la seguridad deben estar correctamente identificadas y perfectamente atribuidas a los departamentos específicos. Se verificará la existencia de un cuerpo normativo de obligado cumplimiento (identificando requisitos normativos) y la difusión del mismo.

  Los incidentes de seguridad detectados deben estar correctamente soportados mediante los análisis y pruebas realizadas. Se deben definir las acciones necesarias para su resolución.
- **Seguridad de la Plataforma**: Es necesario que la seguridad implantada sobre los sistemas que soportan la operativa y almacenan la información sea adecuada, impidiendo el acceso de personal no autorizado. El acceso a los sistemas y el envío de información mediante redes de comunicaciones deberá estar adecuadamente protegido, impidiendo el paso de trasmisiones no autorizadas, así como la intercepción y manipulación de comunicaciones autorizadas.
- **Seguridad de los Usuarios**: Deben existir medidas de control de acceso a los sistemas y a la información de acuerdo con lo establecido en la normativa vigente; interna y externa. Únicamente el personal autorizado debe acceder a los sistemas y efectuar operaciones sobre los datos en base al perfil asignado. El perfil deberá estar alineado con las funciones que desempeñan los usuarios según su puesto, y lo definido por la organización. Los sistemas deben disponer de trazas auditables

que permitan reproducir las acciones que realizaron los usuarios en determinados periodos de tiempo.

- **Seguridad Física:** Se debe garantizar que las instalaciones donde se ubican los sistemas, cuentan con suficientes medidas de protección de acceso físico así como mecanismos adecuados ante posibles desastres.

4. Desarrollo

El desarrollo (construcción, parametrización o implantación) inadecuado de las aplicaciones y plataformas que gestionan la información de la organización pueden provocar pérdidas económicas, deterioros de imagen así como incumplimientos de la legislación vigente, por todo esto, se debe garantizar la existencia de suficientes controles y salvaguardas en el ciclo de vida de desarrollo de las mismas.

- **Gestión de Proyectos de desarrollo:** Deberá existir una metodología definida para los procesos involucrados en el ciclo de vida del desarrollo (gestión de proyectos, metodología), así como controles para el seguimiento de la actividad, herramientas en las que se basa y reportes a la Dirección. Adicionalmente, se debe revisar el proceso de externalización de desarrollos, desde la decisión de factorizar un componente hasta la medida de la calidad del resultado.
- **Ciclo de Vida:** Se revisará el proceso de gestión de la demanda de proyectos, desde la identificación de necesidades en la organización, hasta la formalización de una petición al responsable de desarrollo, valorando la estimación de tiempos y costes de desarrollo. En este sentido, se evaluará el ciclo de desarrollo en lo relativo al cumplimiento de la metodología de construcción de nuevas funcionalidades y de la fase de pruebas realizadas en el mismo.
- **Calidad y satisfacción:** Se evaluarán los mecanismos de medida de la calidad de los desarrollos, así como la satisfacción de los usuarios con los resultados finales y las herramientas utilizadas para tal fin.

5. Aplicaciones

Las aplicaciones son las herramientas que hacen posible a los empleados y clientes la realización de un determinado trabajo u operación, facilitando la introducción y procesamiento de la información de negocio en los sistemas de la organización.

Un conjunto de aplicaciones y sistemas componen la plataforma tecnológica de una organización. La revisión de estas aplicaciones conlleva la evaluación de los siguientes aspectos:

- **Flujos, Riesgos y Controles:** donde se identificarán y evaluarán los procesos principales que conforman la aplicación, los interfaces con las que interactúan y los riesgos principales y medidas de control implantadas para mitigarlos. De igual modo se debe verificar que los procesos están correctamente documentados.
- **Calidad e Integridad de la información:** se comprobará el correcto funcionamiento de los controles que validan los datos existentes en la aplicación (contrastándola con la fuente original de los datos) y se verificará que las características de la información que manejan los procesos se ajustan a los requisitos funcionales definidos.
- **Funcionamiento de los procesos de aplicación:** se verificará el correcto funcionamiento de los procesos de la aplicación (cálculo interno, interfaces, controles implantados, etc.) en base al comportamiento esperado por la organización y la documentación funcional definida.
- **Cobertura funcional y automatización de la plataforma**: se evaluará si el grado de automatización de los procesos de aplicación es suficiente (comprobando la cantidad y complejidad de intervenciones manuales) y si existen procesos no soportados por las aplicaciones; cobertura funcional.
- **Seguridad de los usuarios** : se verificará la existencia de controles que impidan el acceso a personal no autorizado a las aplicaciones, así como la existencia de diferentes perfiles de acceso a la aplicación que aseguren una correcta segregación funcional.

2.4. Metodología general de la auditoría de sistemas de información

Una metodología de auditoría es un conjunto de procedimientos documentados de auditoría diseñados para conseguir los objetivos planificados de la auditoría. Está compuesto de una definición del ámbito de la auditoría, sus objetivos y los programas de trabajo de la auditoría.

Esta metodología debe ser preparada y aprobada por la dirección de auditoría para garantizar la consistencia en los procesos de auditoría. Esta metodología debe ser formalizada y comunicada a todo el personal de auditoría.

Una auditoría, en todos los casos seguirá el programa de trabajo definido en la metodología formal que indicará una serie de fases secuenciales que se seguirán en toda auditoría. De acuerdo a la ISACA [6] una auditoría se debería dividir en las siguientes fases:

1. **Area a auditar** Antes de planificar una auditoría se debe identificar cual será el sujeto de la auditoría. La decisión del área a auditar la tomará la dirección de auditoría basándose en los procesos de *Risk Assesment* que tenga definidos el departamento de auditoría o por encargo concreto.
2. **Objetivo de la auditoría:** Se identifica el propósito de la auditoría. Por ejemplo, un objetivo puede ser determinar si los cambios en el código fuente de las aplicaciones ocurren de acuerdo al procedimiento aprobado en un entorno controlado. Igual que en el punto anterior, esta fase es responsabilidad de dirección de auditoría que tiene la responsabilidad de priorizar unos objetivos frente a otros para optimizar el uso de los recursos.
3. **Alcance de la auditoría:** En la definición del alcance se debe identificar los sistemas específicos, función o unidad de la organización que se incluirá en la revisión. Siguiendo el ejemplo antes mencionado el alcance se podría limitar a la revisión de una sola aplicación, o a un periodo de tiempo limitado.
4. **Planificación de preauditoría:** En esta fase la dirección de auditoría identificará las habilidades y recursos técnicos con los que cuenta. También identificará las fuentes de información para las pruebas o análisis, tales como diagramas de lujo funcionales, políticas, estándares procedimiento o papeles de trabajo de auditorías anteriores. Para finalizar se debe identificar los lugares o instalaciones que se van a auditar.
5. **Procedimientos de auditoría y recolección de datos y realización de pruebas:** En esta fase se debe identificar y seleccionar la aproximación de auditoría para verificar y comprobar los controles de la organización. A continuación se debe preparar una lista del personal que será entrevistado para obtener información relevante sobre lo que se quiere auditar. También se debe identificar y obtener las políticas, los estándares y los procedimientos departamentales para que estos sean

revisados. Para finalizar se desarrollarán o se elegirán las herramientas de auditoría y la metodología para realizar los tests y verificaciones.

6. **Evaluación de los resultados:** Tras la realización de las pruebas y la recolección de los datos, los auditores deben identificar aquellos resultados que suponen una debilidad en el alcance de la auditoría. Para esta evaluación es fundamental que se valore la materialidad de las debilidades ya que el auditor debe estar totalmente seguro de que la solución a estas debilidades es proporcional al efecto potencial de la debilidad si no se toma una acción correctiva. Aquellos resultados que no tengan suficiente materialidad serán reportados dentro del informe de la auditoría pero pueden no suponer una recomendación para el auditado.
7. **Comunicación de los resultados:** Los resultados de la auditoría, en todos los casos, se comunicarán a la dirección de la organización mediante un informe de auditoría en el que se realice una revisión ejecutiva del trabajo realizado, asi como, de las incidencias encontradas. Anexo a este informe se incluirá un pliego de recomendaciones en el que se indique las acciones correctivas que debe tomar el auditado. Este pliego de recomendaciones deberá ser aceptado por el auditado tras la revisión por parte de auditoría del plan de acción a tomar así como del compromiso temporal para la finalización de las acciones correctivas. La realización del informe y del pliego de recomendaciones no será, en ningún caso un proceso unidireccional. Antes de presentar un informe final, sería recomendable que auditoría informara al auditado de las debilidades encontradas y del contenido del informe para así poder negociar con el auditado el contenido del informey los planes de acción.
8. **Seguimiento de las recomendaciones:** Aunque la auditoría haya terminado, el departamento de auditoría mantiene la responsabilidad de realizar un seguimiento continuo de los planes correctivos de las recomendaciones y de la validación y cierre de las mismas. Si la unidad auditada no cumpliera el compromiso adquirido, auditoría deberá reportarlo a la dirección para que se tomen las medidas apropiadas.

A efectos prácticos esta distribución de fases teóricas de la auditoría se simplifica un poco ya que para mejorar la efectividad de las auditorías se potencia la auditoría basada en el riesgo. La auditoría basada en el riesgo se fundamenta en dos procesos:

1. La evaluación del riesgo como input de la planificación de la auditoría

2. La evaluación del riesgo para minimizar el riesgo de auditoría.

Con esta aproximación el auditor se centra en evaluar los controles internos de la organización para asegurar que el riesgo de la organización se detecta y mitiga de forma adecuada.

El proceso de auditoría basada en riesgo se puede ver en el siguiente diagrama

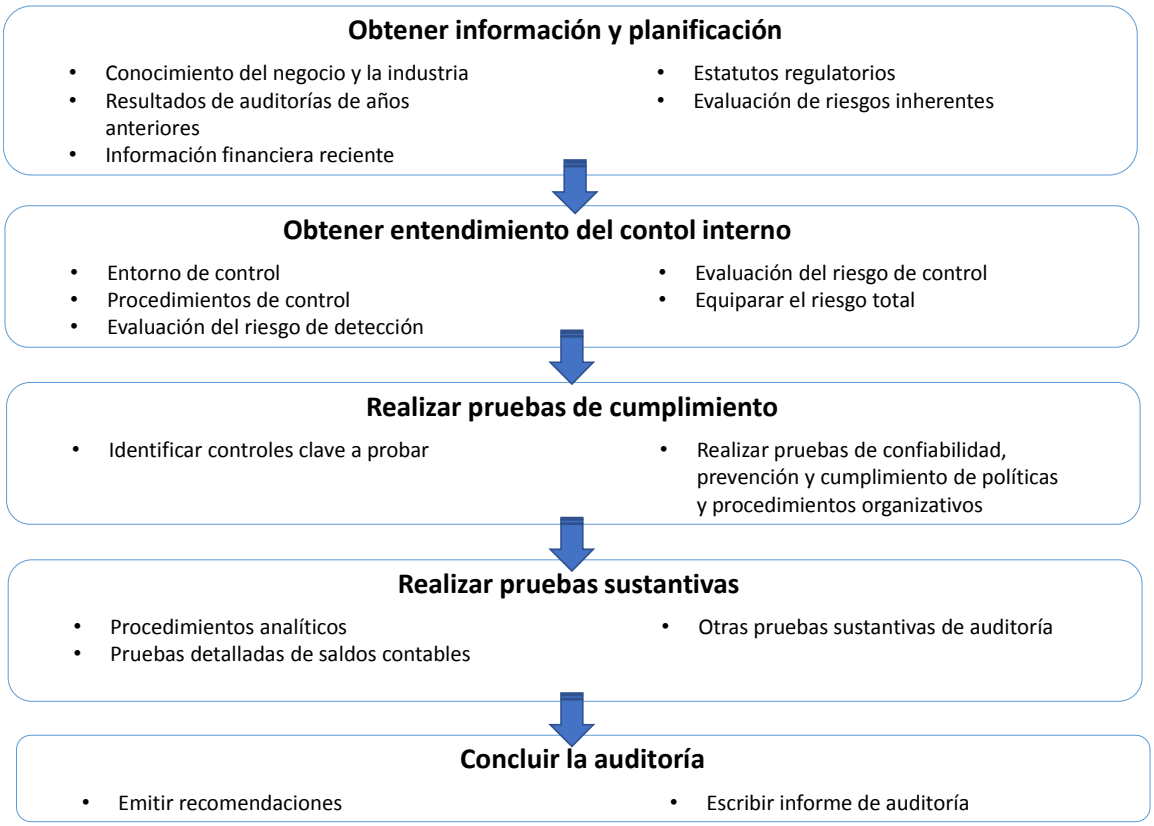


Ilustración 2: Auditoría basada en riesgos

2.5. Evaluación del riesgo

Como se comentó en el punto anterior. El principal input que tiene la dirección de auditoría para tomar la decisión del área, objetivo y alcance de las auditorías en las fases 1, 2 y 3 del proceso de auditoría es la evaluación del riesgo (*Risk Asessment*). Destacamos este proceso ya que la aplicación de auditoría continua que se desarrolla en este proyecto puede ser una herramienta muy útil para que la dirección disponga de información útil, detallada y actualizada en este *Risk Assessment*

Antes de analizar la evaluación de los riesgos deberíamos categorizar los distintos tipos de riesgos que según ISACA [6] [7]se pueden encontrar en una organización.

- **Riesgo inherente:** En lo que atañe al riesgo de auditoría, es el nivel de riesgo o la exposición del proceso/entidad que serán auditados sin tener en cuenta los controles mitigantes que pueda haber en funcionamiento. Los riesgos inherentes existen independientemente de la auditoría y ocurren debido a la naturaleza del negocio.
- **Riesgo de control:** Es el riesgo de que exista un error material que no sea evitado ni detectado a tiempo por el sistema de controles interno. Por ejemplo, el riesgo de control asociado con una revisión manual de los *logs* de sistema puede ser alto porque el volumen de información hace relativamente fácil que una incidencia sea pasada por alto por el técnico encargado de la revisión. El riesgo de control asociado a procedimientos automáticos es relativamente bajo ya que estos funcionan de forma sistemática.
- **Riesgo de detección:** El riesgo de que un error material o fraude no sea detectado por el auditor.

El proceso de evaluación del riesgo debe identificar, cuantificar y priorizar los riesgos frente a los criterios de la organización para la aceptación del riesgo y los objetivos relevantes para la organización. Los resultados de esta evaluación guiarán y determinarán las acciones tomadas por la dirección, las prioridades tomadas para la gestión de los riesgos en seguridad de la información y las prioridades en la implementación de controles que protejan contra estos riesgos.

La evaluación del riesgo no es un proceso puntual sino continuo y debe ser realizada periódicamente para detectar cambios en los sistemas de la información y en los riesgos que afectan a la organización.

Antes de considerar el tratamiento de los riesgos detectados la organización tiene que decidir un criterio para determinar si los riesgos pueden ser aceptados. Por ejemplo si el riesgo es bajo o el coste del tratamiento es mucho más alto que la materialidad del riesgo. Por ejemplo la protección contra terremotos en un centro de datos en un país sin antecedentes sísmicos es un riesgo que puede ser aceptado.

Ante un riesgo detectado una organización tiene las siguientes posibilidades:

- **Mitigación del riesgo:** Aplicación de controles apropiados para reducir los riesgos
- **Aceptación del riesgo:** Eligiendo no tomar ninguna acción, comprobando que la aceptación del riesgo está completamente documentada y de acuerdo a las políticas de la organización.
- **Evitación del riesgo:** Evitando el riesgo al no permitir acciones que provocarían los riesgos
- **Transferencia/Compartición de riesgo:** Transfiriendo los riesgos a otros actores como podrían ser aseguradoras o proveedores.

2.6 Evidencias

Uno de los objetivos de la fase 5 del proceso de auditoría (Procedimientos de auditoría y recolección de datos y realización de pruebas) es la obtención de evidencias que permitan al auditor realizar un juicio sobre el proceso auditado y la realización de pruebas de auditoría.

Una evidencia es cualquier información que utilice el auditor para determinar si la entidad o dato que se está auditando sigue los criterios u objetivos establecidos. Toda conclusión a la que llegue el auditor en el transcurso de la auditoría debe estar soportada por una evidencia relevante.

Las evidencias que el auditor obtenga deben ser confiables, por ello toda evidencia debería seguir los siguientes principios.

- **Independencia del proveedor de la evidencia:** Evidencias de fuentes externas con más confiables que las que provienen de fuentes que tienen algún tipo de interés en la auditoría.
- **Cualificación del individuo que proporciona la evidencia:** Se valorará las cualificaciones y responsabilidades funcionales del individuo que proporciona la información. Esto también afecta a las capacidades del auditor, en caso que el auditor no tenga suficiente conocimiento técnico su juicio para obtener la información puede no ser confiable
- **Objetividad de la evidencia:** Toda evidencia objetiva que provenga de un sistema automático tendrá más relevancia que las que vengan del juicio de una persona, conversaciones o mera observación.

- **Tiempo de disponibilidad de la evidencia:** En sistemas online las evidencias deben estar siempre datadas, ya que la información obtenida a distintas fechas puede no ser válida para evaluar un sistema o la información puede no estar disponible pasado un tiempo

De acuerdo los estándares S6 y S14 de auditoría de ISACA [7] el auditor debe utilizar principalmente las siguientes técnicas para obtener evidencias en el transcurso de una auditoría:

- **Revisión de las estructuras organizacionales:** Se obtendrán organigramas y descripciones funcionales de las divisiones de la empresa para validar la adecuada segregación funcional y el control sobre el desempeño de las funciones de los SI.
- **Revisión de políticas y procedimientos:** Se deberá validar la existencia de políticas y procedimientos que gobiernen las funciones de los SI. También se obtendrán evidencias sobre el entendimiento y cumplimiento de las mismas por parte del personal responsable y la validez de estas políticas y procedimientos comparándolas con las buenas prácticas de la industria. Estas políticas deben estar aprobadas y validadas por la dirección de IT de la organización.
- **Revisión de estándares:** Si los hubiere se revisarán los estándares en vigor en la organización. Estos estándares deben estar aprobados y validadas por la dirección de IT de la organización.
- **Revisión de documentación:** Para el resto de documentación operativa relativa a los SI (Acuerdos de nivel de Servicio con proveedores, Manuales de usuario, Logs de sistema, Planes de Continuidad de Negocio) se analizarán tanto el contenido como la aprobación del documento por el responsable si esta fuera necesaria. Todos estos documentos deberán estar datados para asegurar su vigencia.
- **Entrevista al personal adecuado:** Para obtener evidencias en una entrevista personal el auditor debe apoyarse en un guion de la entrevista para optimizar el tiempo disponible, hacer uso de evidencias ya obtenidas y solicitar evidencias objetivas (Ficheros, capturas de pantalla, documentos) de las afirmaciones del entrevistado.
- **Observación de procesos y del desempeño de los empleados:** Mediante observación directa del proceso de un empleado en su desempeño diario se pueden obtener evidencias sobre la falta de controles operativos o fallos en la operativa. Esta manera de obtener evidencias es vital en las revisiones de seguridad física ya

que el auditor debe obtener evidencia de que los controles funcionan como afirman las políticas de seguridad física de los *datacenters.* También se debe obtener cuando se pueda evidencia documental o fotográfica.

- **Recálculos:** Tomando como partida los datos de entrada de un proceso y su descripción funcional, se modela el cálculo del proceso para reejecutarlo y detectar posibles errores en la aplicación de la organización
- ***Walkthrough:*** Consiste en realizar un seguimiento de un proceso funcional siguiendo todos sus pasos para detectar posibles errores

2.7. Tipos de pruebas de auditoría

En la fase 5 del proceso de auditoría (Procedimientos de auditoría y recolección de datos y realización de pruebas) se plantea que el auditor debe realizar pruebas de auditorías sobre las evidencias obtenidas.

De acuerdo a ISACA [6] hay dos tipos de pruebas que un auditor puede realizar: Pruebas de cumplimiento y pruebas sustantivas

Las pruebas de cumplimiento consisten en la recolección de evidencia con el propósito de comprobar el cumplimiento en una organización de los procedimientos de control. Una prueba de cumplimiento determina si los controles están siendo aplicados de manera que cumplan con las políticas y procedimientos de gestión. Por ejemplo, si queremos comprobar si los controles sobre las librerías de programas de producción limitan la puesta en producción de software no aprobado se seleccionaría una muestra de estos programas para validar si todas las autorizaciones se realizaron de forma apropiada antes de la puesta en producción

Una prueba substantiva valida la integridad de un procesamiento real. Proporciona una evidencia de que la aplicación realiza la función para la que fue diseñada de forma correcta y ayudará a detectar errores en el procesamiento o posibles fraudes. Por ejemplo, una prueba substantiva sería la validación del cálculo de los intereses de la aplicación de crédito hipotecario de una entidad financiera.

2.8. Muestreo

Aunque el uso de técnicas de auditoría asistidas por ordenador (CAAT) ha incrementado la capacidad de realizar pruebas sobre gran cantidad de datos de forma eficiente, en

ocasiones no resulta factible plantear una prueba sobre la totalidad de los datos por limitaciones de espacio, tiempo o personal. En estos casos el auditor debe realizar un muestreo de los datos para seleccionar un subconjunto de los miembros de la población de los datos sobre la que realizar la prueba. Hay dos enfoques principales de muestreo.

- **Muestreo estadístico:** Un método para seleccionar un subconjunto de la población que utilizando cálculos matemáticos y probabilidades selecciona de forma estadística miembros de la población que mantengan las características del conjunto de toda la población.
    - Muestreo aleatorio simple — Asegura que todas las combinaciones de las unidades de muestreo (basadas en la categorización de la muestra) tienen una probabilidad igual de selección.
    - Muestreo sistemático — Se basa en la selección de unidades de muestreo con un intervalo fijo entre selecciones eligiendo de forma aleatoria el comienzo del primer intervalo.
    - Muestreo estratificado aleatorio — Asegura que todas las unidades de muestreo en cada subgrupo tienen una probabilidad conocida distinta de cero de ser seleccionados.

- **Muestreo no-estadístico:** El tamaño de la muestra y los miembros de la misma se determinan en base al juicio del auditor. Estas decisiones se realizan basándose en un criterio subjetivo, lo que no está exento de riesgo. Hay dos métodos principales
    - Muestreo al azar— La muestra se selecciona sin seguir una técnica estructurada para evitar cualquier sesgo o predictibilidad. Este tipo de muestreo no permite extraer una conclusión sobre la población total pero resulta útil en poblaciones que no se puedan caracterizar fácilmente.
    - Muestreo subjetivo— El análisis del auditor es el factor clave en ete tipo de muestreo. Se seleccionan aquellos miembros de la muestra que cumplen un criterio, por ejemplo, aquellos con un saldo mayor de un determinado valor. Este muestreo tampoco tiene unos fundamentos estadísticos así que no se puede extrapolar para toda la población los resultados del test sobre la muestra.

Capítulo 3 Auditoría Continua

La Auditoría Continua es un tipo de auditoría que produce los resultados de la auditoría simultáneamente o en un breve lapso de tiempo desde la ocurrencia de los eventos relevantes. Aunque esta definición es la más comúnmente aceptada, sería más preciso llamar a este tipo de auditoría instantánea en lugar de continua. La confusión se origina debido a que en muchos casos la auditoría instantánea lleva a una producción de resultados siguiendo una alta frecuencia, produciendo un flujo continuo de resultados. Sin embargo, en un proceso de auditoría continua el matiz relevante está en el análisis que se realiza de esos resultados. Aunque la producción de datos sea continua, los resultados de auditoría pueden originarse muy esporádicamente, solamente cuando ocurra un evento relevante.

Una auditoría continua se debe implementar como un proceso automatizado y con acceso instantáneo a los eventos relevantes. La única manera de conseguir cumplir con estos requisitos es implementando el proceso de auditoría continua como un sistema informático online. En este contexto, el sistema online debería tener dos accesos permanentes, uno para que realicen su análisis los auditores y otro para obtener los datos de los auditados.

3.1 Origen

En este punto trataremos de dar una reseña histórica sobre el origen de la auditoría continua. Hay que tener en cuenta que, como todas las técnicas de auditoría, el origen parte de la auditoría contable que históricamente ha sido el tipo de auditoría más relevante en la industria.

De acuerdo a [8] los orígenes de las pruebas de control automatizadas se remontan a la década de 1960 con la instalación e implementación de módulos de auditoría integrados (EAM). Sin embargo, estos módulos eran difíciles de construir y mantener ya que se trataban habitualmente de soluciones ad-hoc y eran utilizados en relativamente pocas organizaciones. A fines de los años 1970, los auditores comenzaron a dejar de lado este enfoque. En la década de 1980, algunos profesionales de auditoría comenzaron a adoptar técnicas de auditoría asistidas por computadora (CAAT, en inglés) para análisis e investigaciones sobre datos automatizables

Simultáneamente, se presentó, por primera vez, la noción de supervisión continua a los auditores en un gran contexto académico. La premisa básica era que el uso de análisis de

datos automáticos permanente ayudaría a que los auditores identifiquen las áreas de mayor riesgo, como punto de partida para determinar sus planes de auditoría. En general, sin embargo, los auditores no estaban preparados para este tipo de enfoque. Carecían de un acceso sencillo a las herramientas de software apropiadas, de pericia y de recursos técnicos para enfrentar desafíos de acceso a los datos y, sobre todo, de la predisposición de las organizaciones para aceptar el compromiso que implicaba la adopción de una metodología y un enfoque de auditoría notablemente distintos.

Durante los años 1990, en la profesión de auditoría a nivel mundial, hubo una adopción generalizada de soluciones de análisis de datos que se consideraron una herramienta crítica para respaldar las pruebas de eficacia de los controles internos. Esta tecnología se empleó para examinar las transacciones en busca de indicadores de incidentes que sucedían porque no se aplicaba un control o porque este no se realizaba correctamente.

También identificaba transacciones que no cumplían con las normas de control. Además, el análisis de datos respaldaba las pruebas de controles que no se evidenciaban en forma directa por medio de los datos transaccionales. Por ejemplo: se podían analizar las tablas de autorización y acceso de planificación de recursos empresariales (ERP, en inglés) para identificar fallas a fin de mantener una separación adecuada de funciones.

No obstante, incluso con esta tecnología como sustento, los procesos de auditoría tradicional a menudo se basaban en muestras representativas en lugar de evaluar toda la población, y los análisis continuaban luego de haber finalizado la actividad de negocio (transacción). Por eso, había más chances de que los problemas de riesgo y control siguieran avanzando y repercutieran negativamente en el desempeño del negocio.

En la actualidad, la proliferación de sistemas de información en el entorno de negocio ofrece a los auditores un acceso más sencillo a una cantidad mayor de información relevante, pero también implica la gestión y la revisión de volúmenes de datos y transacciones mucho más grandes. Además, el ritmo vertiginoso de los negocios requiere una rápida identificación y respuesta a los problemas de control. Regulaciones como la de la Ley Sarbanes-Oxley de Estados Unidos exigen una revelación oportuna de las deficiencias de control y las afirmaciones de la dirección con respecto a la idoneidad del esquema de control.

La auditoría continua permite a los auditores superar los límites de los enfoques de auditoría tradicional y las restricciones de los muestreos, la revisión de informes estándar y las evaluaciones puntuales. Un componente crucial de la auditoría continua es el desarrollo de un modelo de revisión permanente (continuo) de transacciones en el momento exacto, o aproximado, en el que ocurren.

3.2 Enfoque de auditoría continua

Como comentábamos en el capítulo 2, en una organización las auditorías suelen ser un proceso con principio y fin que se puede repetir de forma periódica. Si tomamos como referencia una auditoría concreta sobre un sistema de información fundamental para la organización cabe pensar que todos los años se planificará una auditoría que revise el estado de este sistema.

Tradicionalmente esta aproximación se ha utilizado para optimizar los recursos. Una forma de mantener un control sobre los sistemas minimizando las jornadas/hombre dedicadas a auditarlo es planificar auditorías periódicas con un espaciado adecuado. Con esta planificación siempre se tendrá una imagen más o menos actualizada del estado de los sistemas y en el caso que suceda una incidencia en el sistema, el auditor podrá detectarla en la siguiente auditoría. En la siguiente figura podemos ver el esquema de ete modelo de auditoría.

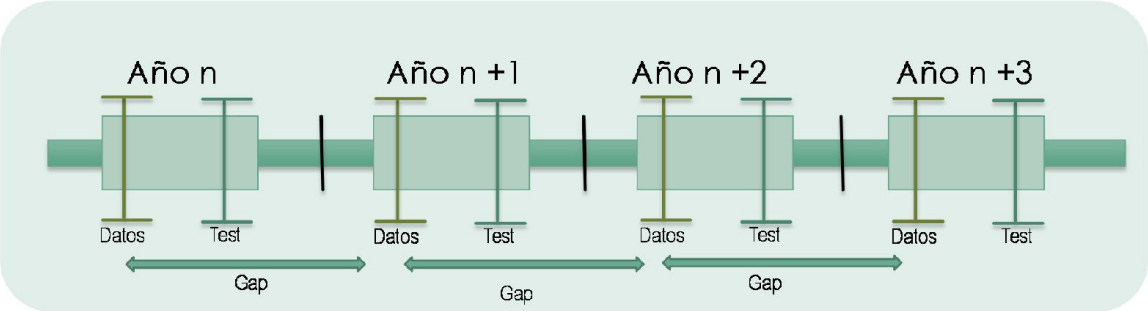


Ilustración 3: Enfoque tradicional de auditoría

Con este enfoque surgen dos problemáticas que la auditoría continua pretende solventar.

- **Vigencia de los datos:** Como se puede ver en la anterior figura, desde que el auditor obtiene los datos hasta que se realiza el test hay un decalaje en el que

el auditor debe analizar los datos que le proporcionó la unidad y desarrollar el test que va a realizar. Si el auditor tuviera disponible en tiempo real los datos de los sistemas a auditar podría realizar las pruebas en la fecha que necesitara y obtener una imagen más fidedigna.

- **Gap entre auditorías:** En la figura podemos ver como desde que se toman las mediciones de una auditoría hasta la siguiente auditoría hay un lapso de tiempo en el que la unidad auditada es, a todos los efectos, una caja negra para auditoría interna. En este lapso de tiempo podrían producirse incidentes, cambios o incluso desaparecer el sistema analizado sin que desde auditoría interna se tenga conocimiento. Incluso el ejemplo que presenta la figura es bastante optimista. Como vimos en el capítulo 2, la planificación de auditorías prioriza unas auditorías sobre otras y se puede dar el caso que el gap entre auditorías sea de varios años.

Para afrontar estos dos riesgos surge el concepto de auditoría continua, que engloba dos actividades principales:

- Evaluación continua de control, destinada a centrarse lo más pronto posible en las deficiencias de control.
- Evaluación continua de riesgos, destinada a destacar los procesos o sistemas que presentan niveles de riesgo más altos de lo esperado.

La frecuencia de la actividad de auditoría continua dependerá del riesgo inherente al proceso o sistema. Además, un enfoque bastante habitual para optimizar recursos es comenzar examinando los controles y las áreas de riesgo clave, y luego ampliar la aplicación de auditoría continua a medida que los auditores ganan experiencia y logran resultados medibles que contribuyan al cumplimiento, la eficacia y la eficiencia operativas y la integridad de los informes financieros

Los resultados de la auditoría y la supervisión continuas (por parte de la dirección) son similares e involucran notificaciones o alertas que indican deficiencias de control o niveles de riesgo más altos que lo deseado. Las notificaciones o alertas se pueden priorizar y, según la gravedad del riesgo o la deficiencia de control, se les pueden entregar a los aseguradores del proceso de negocio o del sistema de aplicación, a los directores

operativos, a los auditores, a los directores financieros e incluso a los organismos de control. La respuesta de la dirección a estas notificaciones puede ser corregir una deficiencia de control y una transacción errónea de inmediato. La respuesta de auditoría a estas advertencias puede abarcar desde una auditoría inmediata del sistema de control identificado hasta la señalización de un área para una futura auditoría.

Para ilustrar el efecto que un enfoque de auditoría continua puede tener en el desempeño de una organización y en los controles que ya hay implantado, tomamos las siguientes ilustraciones de [9]

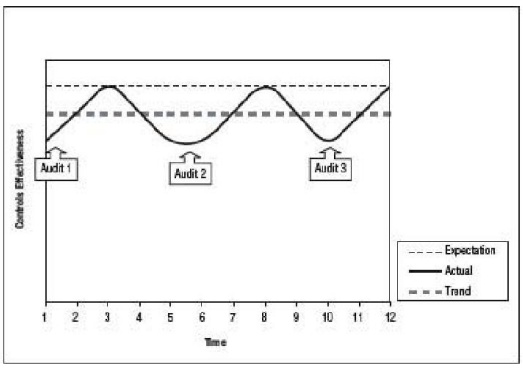


Ilustración 4: Efecto de la auditoría convencional en los controles

En esta gráfica se puede observar cómo tras cada auditoría se produce un *efecto rebote* en el que se observa un mejor desempeño en los controles para implantar las recomendaciones de auditoría. Gradualmente en la organización se va relajando el control hasta la siguiente auditoría. Esto hace que la media real no se acerque al valor deseado.

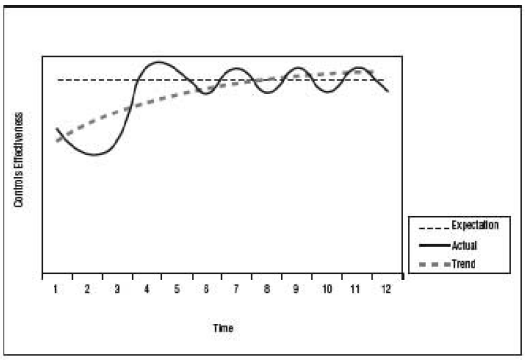


Ilustración 5: Efecto de la auditoría continua en los controles

En un enfoque de auditoría continua el control y feedback continuo hace que las mejoras en el desempeño nunca se pierdan incrementando de forma gradual la media de los resultados de los controles. Esto hace que, por un lado, el resultado obtenido se acerque más al resultado esperado y, por el otro, que no se pierda el control sobre el desempeño de la organización, ya que cualquier desviación será detectada y corregida.

A modo de ejemplo podemos presentar una prueba de auditoría común desde la óptica de la auditoría continua. Las pruebas de auditoría continua de obsolescencia de sistemas operativos pueden emitir notificaciones cuando un porcentaje dado de los servidores está ejecutando un sistema operativo con sistema operativo caducado. La respuesta del auditor puede depender de la criticidad de los sistemas (la respuesta puede ser enviar un mensaje de correo electrónico a los administradores del sistema para recabar una justificación o escalar la incidencia a los gestores de infraestructura IT).

En la siguiente figura podemos ver como plantea la ISACA [10] el proceso de auditoría continua para una empresa típica.

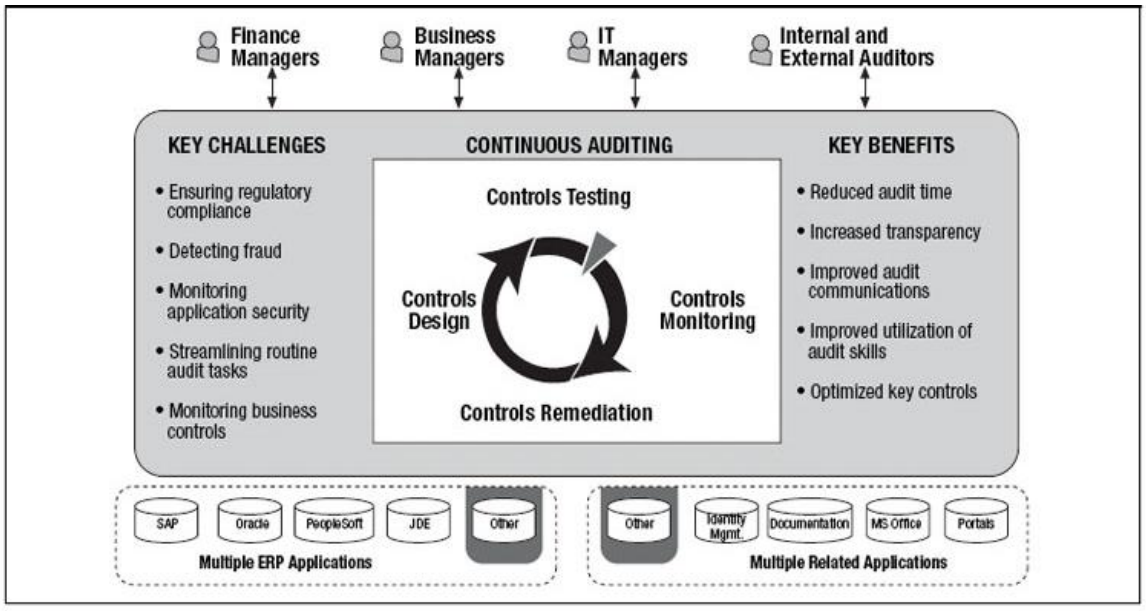


Ilustración 6: La auditoría continua según ISACA

Como vemos en la ilustración ISACA plantea un ciclo de controles continuos sobre datos heterogéneos y una serie de reportes para diversos *stakeholders* de la organización.

3.3 Diferencias entre auditoría continua y monitorización continua

En la literatura sobre la auditoría continua, siempre se aporta un concepto que nos parece importante reseñar. Este es Monitorización Continua. A simple vista la auditoría continua y la monitorización continua parecen procesos muy similares, e incluso procesos paralelos ya que los dos se encargan de obtener información de forma continua y analizarla basándose en las reglas definidas por el propietario del proceso.

La principal diferencia es que la monitorización continúa parte de la dirección de la empresa y genera la información analizada que la dirección necesita, mientras que la auditoría continúa hace lo mismo con el enfoque puesto en el departamento de auditoría. Para comparar ambos enfoques detallaremos las principales características de cada uno.

La monitorización Continua permite a la dirección:

- Medir la efectividad de los controles y detectar los riesgos asociados.
- Mejorar los procesos de negocio y las actividades manteniendo los estándares éticos y de cumplimiento marcados por la dirección
- Ayudar a la toma de decisiones basadas en métricas del riesgo
- Incrementar la eficiencia de los controles y la monitorización usando sistemas informáticos

Como venimos comentando en este capítulo, un sistema de auditoría continua permite:

- Recopilar información de procesos, transacciones y sistemas que permita la realización de las tareas de auditoría.
- Conseguir de forma menos costosa el cumplimiento con las políticas, procedimientos y regulaciones de la organización
- Cambiar las auditorías de un enfoque cíclico y reducido a revisiones continuas y más extensas.
- Cambiar el proceso de planificación a un plan más dinámico basado en los resultados de la auditoría continua.
- Reducción de costes mediante el uso efectivo de sistemas informáticos.

Como se puede ver, los objetivos y motivaciones son complementarios. Si una organización quisiera implementar un sistema de auditoría continua o de monitorización continua, sería

interesante que planteara realizar el esfuerzo de implementar ambas soluciones al mismo tiempo. El único escollo que debe salvar la organización es que se debe tener la absoluta certeza que aunque los datos sean compartidos se mantiene total independencia en los sistemas, de tal manera que el juicio de los auditores y los análisis que realizan son totalmente objetivos y no están, de ninguna manera influenciados por la dirección de la organización.

3.4 Tecnologías utilizadas

Para analizar las tecnologías utilizadas hasta ahora en sistemas de auditoría continua se ha analizado la publicación [11] de la universidad Rutgers de New Jersey que tiene tiene un grupo de investigación centrado en la auditoría continua (*Continous Auditing & Reporting Lab http://raw.rutgers.edu/carlab).*

En este artículo proponen una serie de sistemas utilizados en aplicaciones de auditoría continua

1. **Módulos de auditoría embebidos:** Las aplicaciones de negocio disponen de módulos de auditoría incrustados en el código que permiten establecer alertas configurables y la comunicación con otros sistemas. Estos módulos son un estándar en la industria y podemos encontrarlos desde en los sistemas operativos, bases de datos y aplicaciones empresariales.

2. **Redes neuronales:** Una red neuronal es un Sistema de inteligencia artificial que conecta elementos simples de procesamiento para formar una red. El número de elementos de proceso su interconexión y los pesos de las conexiones determinan la salida para unas entradas especificadas. Las redes neuronales son especialmente efectivas cuando se dispone de un conjunto de ejemplos con salidas conocidas para unas entradas dadas. Las redes neuronales se pueden utilizar como componentes de sistemas expertos de auditoría. En nuestro dominio servirían para modelizar el conocimiento de los auditores y realizar un análisis con inferencia de los datos de la empresa. Sin embargo, en la auditoría de IT no resultan muy útiles, ya que el análisis que hay que realizar sobre los datos es, habitualmente, mucho más simple y la obtención de una serie de reglas no suele ir más allá de unas condiciones sencillas.

3. **Agentes:** Un agente inteligente es un software orientado a objetivos que utiliza internet para obtener los datos que requiere. Se han propuesto sistemas para auditoría continua financiera que realizan búsquedas en internet sobre el dominio a auditar y convierten la información a XBRL para su procesado automático. [12]. Este sistema se puede utilizar como apoyo en queries sobre sistemas online de tal manera que los sistemas de auditoría continua proporcionen información actualizada de distintas fuentes al auditor.

4. **Bases de datos y análisis de datos:** Es la herramienta más simple de agregación de datos que se utiliza en la auditoría continua. Si se quiere automatizar el proceso de auditoría continua os sistemas de auditoría continua deberán conectarse a bases de datos de la organización (Cuando fuera posible a las mismas bases de datos de producción) para obtener la información más actualizada de los sistemas y procesos de negocio. El uso de bases de datos de producción conlleva, sin embargo dos riesgos para el éxito del sistema. Por un lado es difícil conseguir acceso a los datos y por el otro la interpretación y agregación de los datos puede ser realmente complicado y una tarea fuera del alcance del auditor.

5. **Data Warehouses y Data Marts:** Para solventar el problema de análisis e interpretación de las fuentes de datos que indicamos en el punto anterior. Un sistema de auditoría interna puede servirse de los sistemas de *datawarehousing* y *datamart* de las empresas. Un *Data Warehouse* es un repositorio de datos central que se alimenta de los sistemas informáticos operacionales de la empresa. El *Data Mar*t es la capa de acceso a estos datos que permite un análisis y consulta de los mismos. Al disponer de la información agrupada y analizada, el auditor podrá conectar su sistema a los *Data Mart* de la organización para realizar las tareas de auditoría continua.

6. **Data Mining:** El proceso de *Data Mining* o minería de datos está íntimamente relacionado con los repositorios de datos comentados en el punto anterior. El *Data Mining* es el proceso de descubrimiento de patrones en conjuntos grandes de datos. Una vez más, este proceso de análisis de datos y de inferencia de patrones es útil para un sistema de auditoría a distancia general dado que

permitirá obtener alertas tempranas de comportamientos no deseados en sistemas operativos y transaccionales. Desde el punto de vista de un auditor de IS no resulta tan interesante ya que los análisis a realizar son más simples.

7. **Business Intelligence:** Se denomina *Business Intelligence* (BI) a las herramientas automáticas que transforman los datos en bruto en información útil para el análisis de negocio utilizando tecnologías de *reporting*, procesamiento de datos online, *data mining,* etc. Las herramientas de BI no suponen una fuente de datos relevante para un sistema de auditoría continua ya que el auditor intentará, siempre que sea rentable acceder a los datos con las menores transformaciones posibles. Sin embargo lo destacamos aquí ya que un sistema de BI puede ser un componente interesante para el diseño de la interfaz de un sistema de auditoría continua ya que puede facilitar el diseño y procesamiento de los reportes que utilizará el auditor.

3.5 Tareas a realizar en la auditoría continua

Un sistema de auditoría continua que resulte útil para una organización y para un auditor deberá contar con el análisis funcional de un equipo de auditoría que tenga un conocimiento directo de la organización a auditar. El equipo de auditoría intervendrá en las siguientes fases:

1. **Definición de los controles que tendrá el sistema:** Un buen punto de partida sería utilizar como referencia una guía de áreas a auditar como la descrita en el punto 2.3.1 para identificar, al menos, un control para cada uno de los puntos principales y así proporcionar una visión general de todos los dominios de la auditoría de SI. Se deberá definir que se quiere controlar y por qué y proporcionar evidencia documental de estos controles.

2. **Definición de los datos de entrada:** Para cada uno de los controles que se plantean el auditor debe encontrar e identificar una fuente de datos válida lo más automática posible, identificar el grado de procesamiento previo de los datos que y definir una periodicidad con la que se obtienen los datos (Tiempo real / Semanal / Mensual).

3. **Modelización de los controles:** El equipo de auditoría debe definir que procesado se realiza sobre los datos de entrada y qué tipo de test automático se realizará para cumplir el objetivo marcado en el punto 1. En este punto es fundamental que el equipo de auditoría defina unos controles flexibles ya que, cuando se diseñe el sistema se puede encontrar que un determinado control no es implementable por limitaciones de espacio o procesamiento debiéndose pasar a una versión reducida del control.

4. **Definición de alertas y reportes:** Para cada uno de los controles definidos se establecerán unos umbrales de alertas que permitan que la aplicación informe del estado de los controles. Es recomendable disponer de una serie de indicadores de riesgo que en un vistazo simple proporcionen a auditoría el estado global de los sistemas auditados. Aprovechando al información recopilada y los tests realizados es interesante que el sistema permita la generación de reportes dinámicos que auditoría podrá utilizar en pruebas más específicas o como soporte para la emisión de recomendaciones.

5. **Definición de los procedimientos de monitorización continua y seguimiento:** Una vez se tienen los controles y las alertas bien definidos, se debe establecer una metodología clara a seguir para que los empleados a cargo de la solución de auditoría continua respondan de manera adecuada a las alertas del sistema y realicen un seguimiento de las alertas, informando a las unidades interesada para la corrección de las incidencias urgentes en los sistemas. Dentro de este procedimiento se establecerá la relación que tendrá el sistema con el *Risk Assessment* de la organización y la planificación de nuevas auditorías.

3.7 Factores clave en el éxito de un sistema de auditoría continua

No se debe perder de vista que un sistema de auditoría continua es un sistema vivo dentro de una organización y dentro de los sistemas informáticos de una organización está encuadrado dentro de los más difíciles de implementar ya que es un sistema que no está relacionado directamente con el negocio de la organización y es un sistema que no tienen ningún tipo de generación directa de valor.

Por ello basándose en el estudio realizado por el CARLAB de la universidad Rutgers [13] los autores identifican tres factores clave para la adopción de un sistema de auditoría continua [14]

1. **Apoyo de la dirección:** La auditoría continua se ve habitualmente con reticencias desde la dirección ya que es un reto costos y no exento de riesgo. Requiere una inversión considerable y acceso a datos que dependen de distintas partes de la organización. El acceso a estos datos requiere habitualmente la aprobación de la dirección tanto para lograr el acceso a los datos como para conseguir la colaboración de todas las partes en la obtención y procesado periódico de los datos. La participación de la dirección en la adopción de la auditoría distancia es fundamental, si no se ve el proyecto como algo útil no se involucrarán ni presupuestariamente ni en el acceso a los datos.

2. **Competencia de los empleados:** Diseñar e implementar un Sistema de auditoría continuo no es una tarea trivial y tiene una barrera de entrada considerable (tanto en habilidades como en conocimiento tecnológico). Para que los auditores internos puedan monitorizar los controles internos de la organización deben acceder a diversas bases de datos y sistemas de todas las divisiones de la organización. Se necesita que el análisis sea realizado por auditores expertos que puedan identificar los controles clave y desechar aquellos que no merezcan la pena en cuanto al esfuerzo realizado para obtenerlos o procesarlos.

3. **Costes**: El coste inicial de implementación y puesta en marcha de un Sistema de auditoría continua es alto y aunque se pueda justificar con los ahorros a medio y largo plazo que puede proporcionar, no deja de ser un riesgo a la hora de gestionar el proyecto. Los principales beneficiarios de un sistema de auditoría a distancia son los auditores de la organización que consiguen una solución para automatizar sus tareas especialmente diseñada para su organización. No se puede desdeñar el coste de la tecnología implementada y de la formación que se deba dar a los empleados que gestionen y mantengan el uso continuado de la herramienta.

Capítulo 4: Descripción del sistema

En este capítulo procederemos a describir el sistema objeto de este Proyecto de Fin de Carrera detallando el análisis que se ha realizado para su diseño e implementación.

4.1 Introducción

Como comentábamos en el capítulo 3 la auditoría continua es un campo en el que hay mucho margen para la mejora. Especialmente, si centramos el foco en la auditoría de sistemas hay muchas sinergias posibles si se utilizan las herramientas de monitorización y de explotación y análisis de los datos como sistemas fuente de una solución de auditoría continua,

En este capítulo vamos a definir un sistema integrado de auditoría continua que permita obtener los datos de auditoría continua, analizarlos y proporcionar los datos de forma estructurada para su uso por parte de los auditores

4.2 Análisis del sistema

En este apartado se analizará el sistema de auditoría continua que hemos propuesto en este Proyecto de Fin de Carrera. Más allá de hacer un sistema que cubra todas las casuísticas que un Auditor puede encontrarse en el ámbito de la auditoría continua, este análisis estará enfocado en encontrar las tareas que una aplicación de auditoría continua realiza y propondrá un sistema lo suficientemente adaptable para que, con un trabajo de análisis y adaptación al entorno concreto se pudiera implementar en cualquier organización interesada en realizar auditoría continua sobre sus sistemas.

El sistema va a estar enfocado a la auditoría de Sistemas de la Información y será en este dominio en el que se definan una serie de controles. Sin embargo, está fuera del alcance de este análisis la definición de los controles y pruebas de auditoría a implementar ya que la solución propuesta es una solución general. En el apartado de diseño de sistema se propondrá un análisis de ejemplo de un conjunto de pruebas de auditoría y el diseño de las mismas incluyendo datos de entrada, pruebas de auditoría, alertas y umbrales. Para el diseño de las funcionalidades del sistema y la definición de qué se espera de una aplicación de auditoría continua hemos tomado como referencia la Guía de Auditoría Tecnológica Global sobre Auditoría Continua publicada por el Instituto de Auditores Internos[2] (IIA) en la

[2] https://na.theiia.org/Pages/IIAHome.aspx

que se definen las mejores prácticas en el diseño y explotación de un sistema de auditoría continua. [15]

4.2.1 Descripción de las características funcionales

En este apartado se definirán una serie de características funcionales del sistema que darán lugar a los casos de uso y requisitos funcionales del sistema propuesto.

1- **Sistema Online:** El sistema debe ser un sistema online centralizado accesible desde la intranet de la organización. Dado que va a actuar como un repositorio tanto de las pruebas ya calculadas como de los datos que obtenga de los sistemas y procesos, es necesario que el sistema se ejecute en un servidor centralizado conectado a una base de datos, siendo recomendable que se acceda a través de una interfaz web para facilitar la conexión de los usuarios.

2- **Repositorio centralizado de datos procesados de fuentes heterogéneas:** El sistema se conectará a los sistemas que defina el auditor para obtener los datos sobre los que realizar la auditoría continúa. Estos datos provienen de fuentes heterogéneas incluyendo logs de sistema, reportes en Excel, ficheros de configuración, etc. Los datos origen no se almacenarán, pero sí una representación procesada y datada en una base de datos centralizada, sobre la que realizar pruebas de auditoría y obtener informes.

3- **Carga y transformación periódica de datos:** En un mundo ideal el sistema se conectaría directamente a los sistemas de producción para obtener la información. Sin embargo debemos estar preparados para el momento en que la política corporativa se cruce con nuestras buenas intenciones. El sistema debe proporcionar un sistema para la carga periódica de datos manual o automática. Este sistema se encargará de validar, transformar y cargar los datos en el formato utilizado por el sistema, en una base de datos centralizada y manteniendo información sobre la fecha de los datos.

4- **Realización de pruebas de auditoría continua:** Para cada uno de los puntos de control definidos por los auditores de la organización de los que se disponga de datos periódicos el sistema debe permitir la definición de una o varias pruebas de

auditoría y debe calcular estas pruebas con la periodicidad que el auditor haya definido.

5- **Definición de KRIs:** Para el acceso a los resultados de las pruebas de auditoría el sistema debe permitir la creación de un *Key Risk Indicator* o KRI. Este KRI es una representación visual de la prueba de auditoría para que el auditor, o la dirección pueda identificar de forma fácil el cumplimiento de la prueba de auditoría. Estos KRIs deben ser fácilmente modificables a través de la interfaz de la aplicación.

6- **Definición de umbrales de KRIs:** Para estipular el cumplimiento en los KRIs y proporcionar granularidad en la información que se muestra, el sistema debe permitir la creación de umbrales como valores puntuales o rangos. Así el cumplimiento de los KRIs se podrá informar en una escala definida por el auditor.

7- **Reportes de KRIs –** Los KRIs definidos por el auditor se mostrarán en un reporte integrado que permita una vista por dominio de auditoría y que muestre el valor del KRI, la valoración del mismo basándose en los umbrales definidos y una archivo histórico de los valores de cada KRI.

8- **Alertas relativas a KRIs –** El sistema debe permitir la definición de alertas configurables que informen a los responsables cuando un KRI supere un umbral definido.

9- **Interfaz de reportes –** Para el acceso detallado a las pruebas de auditoría o a los datos de origen, el sistema debe incorporar una interfaz de *reporting* configurable, que permita la exportación de los datos a los formatos más habituales.

4.2.2 Restricciones del sistema

En el desarrollo de este sistema planteamos dos restricciones fundamentales para garantizar el éxito de la implementación del sistema. Si en algún momento planteáramos un proyecto como este a una empresa se debería tener en cuenta la aversión al riesgo de la dirección de las organizaciones, las limitaciones presupuestarias y los recursos técnicos de los que disponga la empresa. Para minimizar estos tres factores debemos adoptar un

compromiso y diseñar un sistema que sea lo más barato, lo más estandarizado y lo más adaptable posible. Es por ello que planteamos las siguientes dos restricciones:

- **Uso de soluciones software estandarizadas (*Off the Shelf*)** – Para evitar complejidades limitaremos el desarrollo de soluciones ad-hoc. Todos los módulos que se utilicen en el sistema serán soluciones del mercado que requieran una ligera adaptación para cumplir las tareas que proponemos. En la selección de estos componentes se ponderará positivamente las soluciones de coste cero.

- **Uso de soluciones Open Source** – Como asumimos que el personal encargado del desarrollo y mantenimiento de la solución no tiene por qué ser expertos en ninguno de los módulos y queremos minimizar el coste del desarrollo el apoyo técnico de una comunidad Open Source puede significar la diferencia entre el fracaso y el éxito de la solución. El uso de soluciones Open Source también puede facilitar modificaciones en los módulos que ayuden a adaptar la solución a nuestro dominio.

4.2.3 Especificación de casos de uso

En este apartado se definen una serie de casos de usos que ejemplifiquen la interacción del usuario con el sistema que ayuden a entender las necesidades que nuestro sistema debe cubrir.

4.2.3.1 Descripción de los actores

El sistema de auditoría a distancia, una vez diseñado e implementado, será utilizado por un departamento de auditoría ya sea interno o externo. Por ello los actores elegidos para el caso de uso son los diferentes niveles funcionales dentro de un departamento de auditoría

AC 001- ***Auditor Jr***. Encargado de cargar datos en el sistema. Como nivel más bajo dentro del departamento de auditoría se encarga de la parte más rutinaria, la carga de datos y resolución de incidencias.

AC 002- ***Auditor Manager*** que consulta KRIs. El Auditor Manager tendrá contacto con la aplicación para obtener información ya procesada. Necesita una visión ejecutiva que facilite la toma de decisiones.

AC 003- ***Auditor Senior*** que extrae reportes para realizar pruebas de auditoría. El Auditor Senior utilizará la aplicación como repositorio de información procesada y esperará encontrar informes que le sirvan como evidencias para pruebas de auditoría más avanzadas que vaya a realizar.

4.2.3.2 Descripción de los atributos de los casos de uso

Para la realización de la descripción textual de los distintos casos de uso, se han seleccionado una serie de atributos que describen cada uno de los casos de uso. A continuación se realiza una descripción del significado de cada uno de los atributos utilizados para la descripción de los casos de uso.

- Código: Identificación unívoca abreviada del caso de uso, se construye mediante CU seguido de un - y de tres dígitos. Por ejemplo CU-001.
- Nombre: Identificación extendida del caso de uso.
- Actores: Conjunto de entidades que interactúan con el caso de uso. El caso de uso representa una funcionalidad demandada por un actor.
- Descripción: Se realiza una descripción básica de la funcionalidad o funcionalidades del caso de uso.
- Precondiciones y postcondiciones: Se realiza una descripción de las condiciones que deben cumplirse para poder realizar una operación, y el estado en el que queda el sistema tras realizar una operación.
- Escenario: Se realiza una descripción básica de las acciones que se ejecutaran paso a paso en el caso de uso.

3.2.3.3 Descripción textual de los casos de uso

Caso de uso	
Código	**CU-001**
Nombre	Carga de datos de una fichero Excel o CSV
Actores	**AC-001** Auditor Jr
Descripción	Una operativa habitual es incorporar periódicamente los ficheros fuente a la aplicación. El auditor recibirá los datos de la unidad en formato Excel o CSV para cargarlos a través de la interfaz de carga de la aplicación. Una vez subido a la aplicación, se pueden realizar una serie de comprobaciones sobre el fichero para su validación y finalmente el fichero se cargará en la base de datos quedando disponible para que la aplicación siga realizando los cálculos que tiene definidos.
Precondiciones	El auditor ha recibido un fichero para cargar en el sistema. El auditor tiene permiso en el sistema para la carga de datos.
Poscondiciones	Si no hubo errores en la carga, el fichero se incorpora a la base de datos, si los hubo, se informa al auditor y no se escribe nada en la base de datos.
Escenario	1 – El auditor entra en la aplicación y accede a la carga de datos 2 – El auditor selecciona el fichero que va a cargar 3 – El sistema realiza comprobaciones sobre el fichero 4 – El sistema carga el fichero en la base de datos 5 – El auditor recibe confirmación de la carga con un informe sobre los posibles errores

Tabla 5 - Caso de uso CU-001

Caso de uso	
Código	**CU-002**
Nombre	Consulta de un informe KRI
Actores	**AC-002** Auditor Manager **AC-003** Auditor Senior
Descripción	La operativa habitual de análisis de las pruebas de auditoría continua será el acceso al informe de KRIs en el que el auditor podrá acceder a los valores de cada KRI clasificados por dominio de la auditoría. Cada KRI debe incluir su valor así como la valoración del KRI basada en los umbrales definidos.
Precondiciones	Se ha definido un informe KRI en el sistema. El auditor tiene permiso para consultar el informe Se han cargado los datos de cada uno de los KRIs
Poscondiciones	Se obtiene por pantalla un informe KRI
Escenario	1 – El auditor entra en el sistema y solicita un informe KRI 2 – La aplicación calcula el informe y le devuelve un listado de los KRIs con su valor y una clasificación basada en los umbrales definidios

Tabla 6 - Caso de uso CU-002

Caso de uso	
Código	**CU-003**
Nombre	Descarga del reporte de un indicador
Actores	**AC-003** Auditor Senior **AC-002** Auditor Manager
Descripción	Para el acceso a los resultados detallados de las pruebas de auditor así como el acceso a los datos de origen el sistema debe proporcionar una interfaz simplificada de reporting. Esta interfaz proporcionará los informes por pantalla, pero también permitirá la exportación de los informes en los formatos habituales.
Precondiciones	Se ha definido un reporte detallad en la aplicación Se han cargado los datos del reporte El auditor tiene permiso para ejecutar el reporte
Poscondiciones	Se obtiene por pantalla el reporte solicitado
Escenario	1– El auditor entra en el sistema y solicita un informe de los ya diseñados 2– La aplicación calcula el informe y devuelve los datos procesados por pantalla. El usuario puede descargarlos a su equipo en formato Excel o CSV.

Tabla 7 - Caso de uso CU-003

4.2.4 Especificación de requisitos

En este apartado se presentará una descripción tabulada de los requisitos del sistema.

4.2.4.1 Descripción de los atributos de los requisitos

Para la realización de la descripción textual de los distintos requisitos que han sido identificados, se han seleccionado una serie de atributos que describen cada uno de los requisitos. A continuación se realiza una descripción del significado de cada uno de los atributos utilizados para su descripción:

- Código: Identificación unívoca abreviada del requisito, se construye mediante el código del requisito seguido de un - y de tres dígitos. Los requisitos serán divididos en funciones y no funcionales y sus códigos son RF para los requisitos funciones y RNF para los requisitos no funcionales. Los requisitos se han dividido también en requisitos del módulo de carga de datos y requisitos del módulo de auditoría (RFC y RFA respectivamente).
- Nombre: Identificación extendida del requisito.
- Descripción: Se realiza una descripción básica del requisito que ha sido identificado.
- Fuente: Indica a través de que fuente ha sido identificado el requisitos. Normalmente este valor se corresponderá con uno o varios códigos de los casos de uso.
- Necesidad: Determina el grado de implementación del requisito. Los valores que puede tomar este atributo son los siguientes:
  - Esencial: El requisito tiene que ser implementado.
  - Deseable: Es preferible implementar el requisito, pero no es obligatorio.
  - Opcional: El requisito se podrá implementar, pero no es importante ni obligatorio.
- Prioridad: Define la importancia del requisito, de forma que permita definir el orden en el cual serán incluido en el proceso de diseño y el orden de implementación. Los valores que puede tomar este atributo son los siguientes:
  - Alta: El requisito debe ser implementado en las fases iniciales del desarrollo.
  - Media: El requisito debe ser implementado una vez que hayan sido implementados los requisitos de prioridad alta.

- Baja: El requisito debe ser implementados en las fases finales del desarrollo. Estos requisitos no influirán en el correcto funcionamiento del sistema.

- Estabilidad: Define la estabilidad del requisitos durante la vida útil del software. Esto implica si el requisito podrá ser o no modificado durante el ciclo del vida. Los valores que puede tomar este atributo son los siguientes:
  - Estable: El requisito no puede variar durante el ciclo de vida del sistema.
  - Inestable: El requisito puede variar a lo largo de la ciclo de vida del sistema.
- Verificabilidad: Define el grado de verificabilidad de un requisito, es decir indica en qué grado es posible comprobar que el requisito se ha incorporado en el sistema desarrollado. Los valores que puede tomar este atributo son los siguientes:
  - Alta: Se puede verificar que el requisito ha sido implementado en el sistema. Este tipo de requisitos se corresponden con las funcionalidades básicas del sistema.
  - Media: Se puede verificar que el requisito ha sido implementado en el sistema. Pero requiere de una comprobación compleja o del código fuente del sistema.
  - Baja: Es difícil verificar si el requisito ha sido implementado en el sistema o en algunos casos no es posible.

4.2.4.2 Requisitos funcionales del módulo de carga de datos

Requisito del sistema			
Código	**RFC-001**	**Fuente**	**CU-001**
Nombre	Control de acceso basado en usuarios y roles		
Descripción	El módulo de carga de datos debe proporcionar un método para la autenticación de usuarios con ID de usuario y password. Adicionalmente, las cuentas de usuario tendrán permisos modulares basados en roles		
Necesidad	Deseable	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 8 - Requisito del Sistema RFC-001

Requisito del sistema			
Código	**RFC-002**	**Fuente**	**CU-001**
Nombre	Definición de ficheros de entrada		
Descripción	El módulo de carga de datos debe permitir el alta y definición de ficheros de entrada. En esta definición se estipulará un nombre de fichero, un formato y una descripción de las diferentes columnas del fichero..		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 9 - Requisito del sistema RFC-002

Requisito del sistema			
Código	**RFC-003**	**Fuente**	**CU-001**
Nombre	Programación de transformación de ficheros		
Descripción	El módulo de carga de datos permitirá la programación de las operaciones lógicas de transformación de ficheros en un lenguaje de alto nivel. Deberá permitir tanto transformaciones independientes como combinaciones y filtrados entre varios ficheros de entrada.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 10 - Requisito del Sistema RFC-003

Requisito del sistema			
Código	**RFC-004**	**Fuente**	**CU-001**
Nombre	Programación de carga de ficheros en BD		
Descripción	El módulo de carga de datos deberá permitir la creación de tablas en la Base de Datos y la carga de los ficheros transformados en esta base de datos para que sean usados por el módulo de auditoría		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 11 - Requisito del Sistema RFC-004

Requisito del sistema			
Código	**RFC-005**	**Fuente**	**CU-001**
Nombre	Validación de ficheros		
Descripción	El módulo de carga de datos deberá permitir la definición de condiciones lógicas de validación de ficheros incluyendo: • Formato de fichero • Número de columnas • Tipos de datos en columnas • Rangos válidos en columnas Con estas condiciones definidas, se realizarán pruebas de validación **antes de la carga de datos.**		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 12 - Requisito del sistema RFC-005

Requisito del sistema			
Código	**RFC-006**	**Fuente**	**CU-001**
Nombre	Informes sobre carga de ficheros		
Descripción	El módulo de carga de datos generará informes del resultado de la carga tras un intento de carga, siendo esta exitosa o no. El informe de carga contendrá información suficiente para que el usuario pueda depurar el fichero de carga para conseguir una carga exitosa.		
Necesidad	Esencial	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 13 - Requisito del sistema RFC-006

Requisito del sistema			
Código	**RFC-007**	**Fuente**	**CU-001**
Nombre	Alta, Baja y Modificación de transformaciones de ficheros		
Descripción	El módulo de carga de datos permitirá realizar operaciones de mantenimiento sobre la transformación de ficheros incluyendo las operaciones de Alta, Baja y Modificación.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 14 - Requisito del sistema RFC-007

Requisito del sistema			
Código	**RFC-008**	**Fuente**	**CU-001**
Nombre	Almacenar periodicidad de carga de ficheros		
Descripción	El módulo de carga de datos incorporará un parámetro en cada fichero definido con la periodicidad esperada de actualización de ficheros.		
Necesidad	Deseable	**Prioridad**	Baja
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 15 - Requisito del sistema RFC-008

Requisito del sistema			
Código	**RFC-009**	**Fuente**	**CU-001**
Nombre	Alertas de periodicidad de carga de ficheros		
Descripción	Para cada fichero con periodicidad definida el módulo de carga de datos incorporará un sistema automático de alertas que avise de que la carga de datos está pendiente.		
Necesidad	Deseable	**Prioridad**	Baja
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 16 - Requisito del sistema RFC-009

Requisito del sistema			
Código	**RFC-010**	**Fuente**	**CU-001**
Nombre	Mantener documentación sobre transformación y carga de ficheros		
Descripción	El módulo de carga de datos debería proporcionar un mecanismo para asociar documentación relevante que describa los ficheros esperados y los procedimientos de transformación y carga que se realizarán con ellos.		
Necesidad	Deseable	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 17 - Requisito del sistema RFC-010

4.2.4.3 Requisitos funcionales del módulo de auditoría

Requisito del sistema			
Código	**RFA-001**	**Fuente**	**CU-002**
Nombre	Control de acceso basado en usuarios y roles		
Descripción	El módulo de auditoría debe proporcionar un método para la autenticación de usuarios con ID de usuario y password. Adicionalmente, las cuentas de usuario tendrán permisos modulares basados en roles		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 18 - Requisito del sistema RFA-001

Requisito del sistema			
Código	**RFA-002**	**Fuente**	**CU-002**
Nombre	Alta, Baja y modificación de usuarios		
Descripción	El módulo de auditoría debe permitir la gestión de usuarios con las operaciones de mantenimiento habituales (Alta, Baja y Modificación). La ejecución de estas operaciones debe estar restringida a usuarios con un rol *Administrador* en la aplicación.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 19 - Requisito del sistema RFA-002

Requisito del sistema			
Código	**RFA-003**	**Fuente**	**CU-002**
Nombre	Definición de orígenes de datos		
Descripción	El módulo de auditoría debe incorporar un mecanismo de conexión a Bases de Datos que permita definir uno o varios orígenes de datos para obtener la información cargada por el módulo de carga de datos que se utilizará para realizar pruebas de auditoría.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 20 - Requisito del sistema RFA-003

Requisito del sistema			
Código	**RFA-004**	**Fuente**	**CU-002**
Nombre	Definición de pruebas de auditoría		
Descripción	El módulo de auditoría permitirá la definición lógica de pruebas de auditoría, el origen de datos de las pruebas, así como la programación de estas pruebas mediante un lenguaje de alto nivel.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 21 - Requisito del sistema RFA-004

Requisito del sistema			
Código	**RFA-005**	**Fuente**	**CU-002**
Nombre	Definición de umbrales de pruebas de auditorías		
Descripción	Para las pruebas definidas, el módulo de auditoría debe permitir la definición de umbrales puntuales o basados en rangos para poder calcular la valoración de cada prueba de auditoría.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 22 - Requisito del sistema RFA-005

Requisito del sistema			
Código	**RFA-006**	**Fuente**	**CU-002**
Nombre	Definición de KRIs		
Descripción	El módulo de auditoría debe permitir asociar una prueba de auditoría a unos umbrales definidos para constituir un *Key Risk Indicator* (KRI) que permita valorar el cumplimiento de las pruebas de auditoría.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 23 - Requisito del sistema RFA-006

Requisito del sistema			
Código	**RFA-007**	**Fuente**	**CU-002**
Nombre	Definición de informe de KRIs		
Descripción	El módulo de auditoría debe proporcionar un informe gráfico en el que se pueda consultar los KRIs definidos en una estructura jerárquica, el valor de cada uno de los KRIs y una valoración basada en los umbrales codificada con una escala de colores.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 24 - Requisito del sistema RFA-007

Requisito del sistema			
Código	**RFA-008**	**Fuente**	**CU-002**
Nombre	Cálculo periódico de KRIs		
Descripción	El módulo de auditoría debe permitir programar un cálculo periódico de los KRIs definidos sobre los últimos datos presentes en la Base de datos.		
Necesidad	Esencial	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 25 - Requisito del sistema RFA-008

Requisito del sistema			
Código	**RFA-009**	**Fuente**	**CU-002**
Nombre	Valoración de KRIs		
Descripción	El módulo de auditoría debe permitir valorar el resultado de los KRIs asignando un código de colores en función de los umbrales definidos.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 26 - Requisito del sistema RFA-009

Requisito del sistema			
Código	**RFA-010**	**Fuente**	**CU-002**
Nombre	Almacenamiento de histórico de cálculo de KRIs		
Descripción	El módulo de auditoría debe almacenar un histórico sobre la valoración de los KRIs definidos. Este histórico se debe de poder consultar tanto de forma gráfica (Con una gráfica de tendencia) como numérica.		
Necesidad	Esencial	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 27 - Requisito del sistema RFA-010

Requisito del sistema			
Código	**RFA-011**	**Fuente**	**CU-003**
Nombre	Definición de informes visuales		
Descripción	El módulo de auditoría debe permitir la creación de informes visuales para proporcionar acceso a los datos en pruebas de auditoría. Los informes deben ser configurables para poder adaptarlos al estilo de la organización.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 28 - Requisito del sistema RFA-011

Requisito del sistema			
Código	**RFA-012**	**Fuente**	**CU-003**
Nombre	Cálculo de informes bajo demanda		
Descripción	Los informes definidos deberán poder calcularse con los datos más recientes de la base de datos y no se calcularán salvo que el auditor necesite consultarlos.		
Necesidad	Esencial	**Prioridad**	Alta
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 29 - Requisito del sistema RFA-011

Requisito del sistema			
Código	**RFA-013**	**Fuente**	**CU-003**
Nombre	Control de acceso a informes basado en roles		
Descripción	Para evitar acceso a datos confidenciales, la ejecución y consulta de los informes se restringirá a roles limitados que serán asignados a usuarios		
Necesidad	Deseable	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 30 - Requisito del sistema RFA-013

Requisito del sistema			
Código	**RFA-014**	**Fuente**	**CU-003**
Nombre	Exportar reportes en formatos estándar		
Descripción	Los reportes que se consulten a través de la aplicación se deberán poder exportar en formatos estándar para su distribución en la organización o para ser utilizados como fuente de datos para pruebas de auditoría. Los formatos requeridos son: • Excel - XLS/XLSX • Portable Document Format – PDF • Comma Separated Values – CSV		
Necesidad	Esencial	**Prioridad**	Media
Estabilidad	Estable	**Verificabilidad**	Alta

Tabla 31 - Requisito del sistema RFA-014

4.2.4.4 Requisitos no funcionales

Requisito del sistema			
Código	**RNF-001**	**Fuente**	
Nombre	Uso de componentes Off the Shelf		
Descripción	Para minimizar el coste de implementación y mantenimiento de la solución se elegirán componentes que ya existan en el mercado, intentando que todo el trabajo de adaptación sea parametrización de los componentes.		
Necesidad	Deseable	**Prioridad**	Alta
Estabilidad	Inestable	**Verificabilidad**	Media

Tabla 32 - Requisito del sistema RNF-001

Requisito del sistema			
Código	**RNF-002**	**Fuente**	
Nombre	Uso de componentes *Open Source*		
Descripción	Para minimizar el coste de implementación y mantenimiento de la solución se elegirán componentes *Open Source*. Asimismo las herramientas que se utilicen en el diseño y desarrollo del sistema serán también *Open Source*.		
Necesidad	Deseable	**Prioridad**	Baja
Estabilidad	Inestable	**Verificabilidad**	Media

Tabla 33 - Requisito del sistema RNF-002

4.3 Diseño del sistema

En este apartado comentamos el diseño del sistema desde un punto de vista abstracto hasta la elección de los componentes concretos que se implantarán en la aplicación.

4.3.1 Arquitectura del sistema

Basándonos en los requisitos planteados, especialmente los no funcionales queda claro que la solución que implementemos debe tener un repositorio centralizado en el que se almacenen tanto los datos de entrada como los resultados de las pruebas. Este repositorio debería ser una base de datos que permita acceso concurrente y la ejecución de consultas estructuradas.

Para el acceso a la funcionalidad de la aplicación se diferencian claramente dos módulos funcionales, un módulo de auditoría que se encargue de los cálculos de pruebas de auditoría y de la generación de informes y un módulo de carga que admita ficheros con los datos en bruto, los valide y los cargue en la base de datos.

En el siguiente diagrama podemos ver una representación lógica de la arquitectura propuesta.

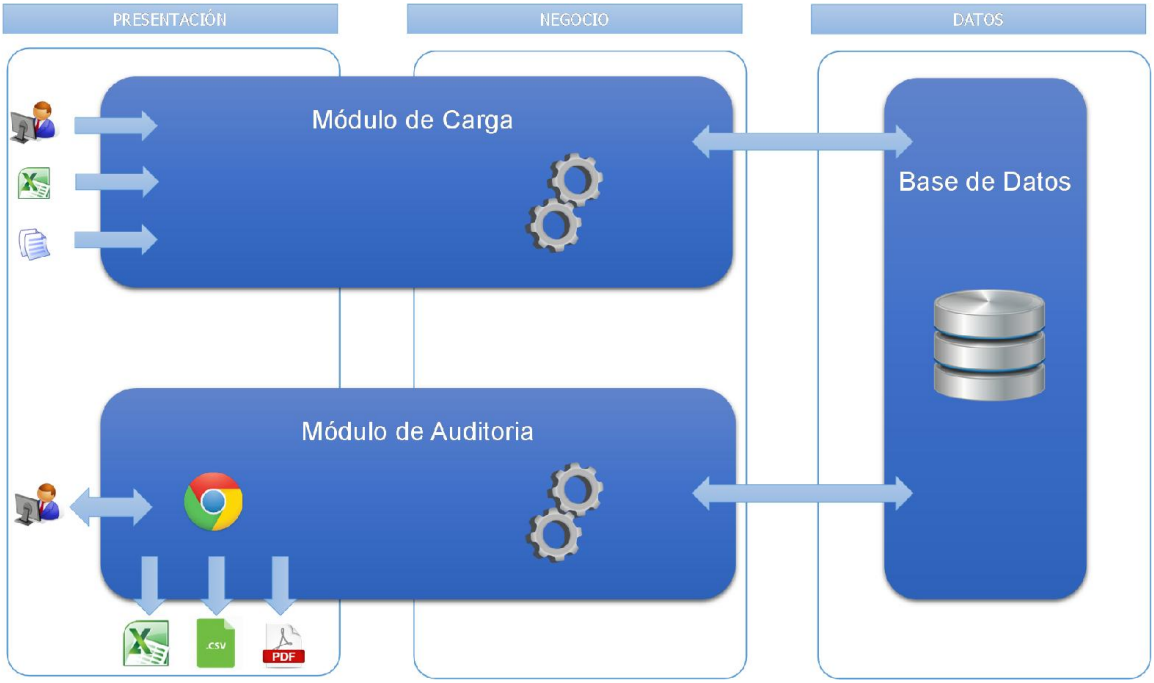


Ilustración 7: Arquitectura lógica del sistema

Como se puede ver en el diagrama, hemos introducido alguna de las limitaciones del sistema que nos ayudarán a elegir los componentes más adelante. Por ejemplo, en el diseño general de arquitectura ya se define una capa de presentación web para el módulo de auditoría y una serie de formatos de salida para los informes. Por lo demás el diseño aquí presentado es muy general y será cuando seleccionemos los componentes cuando se presentará un diagrama detallado de la arquitectura del sistema.

4.3.2 Descripción de componentes

En este apartado se proporcionará una descripción detallada de los tres componentes principales del sistema (Módulo de Auditoría, Módulo de Carga y Base de Datos). Se presentará el análisis realizado para elegir un componente u otro así como consideraciones de diseño de cada uno de los componentes.

4.3.2.1 Módulo de Auditoría

Para cumplir con los requisitos funcionales explicitados anteriormente debemos seleccionar un módulo que realice las funciones de auditoría continua. Este módulo debe poder realizar los cálculos y pruebas estipulados sobre los datos cargados en la base de datos así como mostrar estos resultados de una manera visual y configurable.

Entre los sistemas que encontramos en el mercado la solución integrada que mejor se adapta a nuestro problema sería utilizar una suite de Business Intelligence (BI). Se conoce como Business Intelligence al conjunto de técnicas y herramientas para la transformación de datos brutos en información útil para el análisis del negocio. Traducido a nuestro dominio la suite de BI nos permitirá definir una serie de pruebas heterogéneas de auditoría sobre nuestro repositorio central de datos, integrarlas en un reporte visual y proporcionar reportes de detalle sobre las pruebas de auditoría y los datos de origen. Adicionalmente una suite de BI puede proporcionar otras funcionalidades que cubren algunos de nuestros requisitos: Acceso a uno o varios repositorios de datos como fuente

- Definición de reportes de complejidad variable
- Control integrado de usuarios y roles
- Generación de informes KRI

Las soluciones que hemos evaluado son las siguientes

Jaspersoft

https://www.jaspersoft.com/es

Jaspersoft es una herramienta *Open Source* con un modelo de negocio *Open Core. Jaspersoft* tiene una edición *community* gratuita y versiones con coste con mayores funcionalidades (*Reporting, AWS, Professional y Enterprise*). *Jaspersoft* se distribuye bajo la licencia GNU *Affero General Public License 3.0* (AGPLv3)[3]

Pentaho

http://www.pentaho.com/

Pentaho sigue un modelo de negocio de *Open Core*. Proporcionan dos ediciones diferentes de *Pentaho Business Analytics*: una edición *community* y una edición *enterprise*. La edición *enterprise* tiene coste de licencia y sigue un modelo de subscripción. Hay tres variantes de la edición *enterprise*: Basic, Professional, and Enterprise. La edición *community* es un producto gratuito *Open Source* licenciado bajo la licencia *GNU General*

[3] http://www.gnu.org/licenses/agpl-3.0.html

Public License version 2.0 (GPLv2), [4]*GNU Lesser General Public License version 2.0* (LGPLv2), y *Mozilla Public License 1.1* (MPL 1.1).[5]

SpagoBI

http://www.spagobi.org/

La suite *SpagoBI* es un software gratis *Open Source* distribuido bajo la licencia *Mozilla Public License* v 2.0 sin la cláusula *"Incompatible With Secondary Licenses"*[6]. (Hasta la versión 3.4 *SpagoBI* se distribuía bajo la licencia GNU LGPL[7]).

En [16] se realiza una comparativa entre soluciones Open Source de BI. Como podemos ver en la siguiente tabla, la propuesta tecnológica de los líderes del mercado es bastante similar ofreciendo todas ellas unas prestaciones parecidas

Módulos	JasperSoft	Pentaho	SpagoBI
Servidor de Aplicación	JBoss	JBoss	Tomcat
Autenticación y perfilado de usuarios	Acegi	Acegi	Integrado en el eXoPortal
Colaboración	-	-	Dossier d
Dashboard	JFreeChart	JFreeChart	Openlaszlo
Data Mining	-	Weka	Weka
DBMS	MySQL, Oracle, SQL Server, PostgreSQL, etc.	MySQL, Oracle, SQL Server, PostgreSQL, etc.	MySQL, Oracle, SQL Server, PostgreSQL, etc.
ETL	JasperETL	Pentaho Data Integration	Talend Open Studio
Geo-referencing	Google Maps	Google Maps	GEO
Job Scheduler	Quartz	Quartz	Quartz
OLAP	Mondrian&Jpivot	Mondrian&Jpivot	Mondrian&Jpivot
Portal	Liferay	Jboss Portal	ExoPortal, Liferay
Query by Example	-	-	Hibernate
Reporting	JasperReport	Pentaho Report Designer, JasperReport, BIRT	JasperReport, BIRT
Single Sign On	Acegi	CAS	CAS

[4] http://www.gnu.org/licenses/gpl-2.0.html
[5] https://www.mozilla.org/MPL/1.1/
[6] https://www.mozilla.org/MPL/2.0/
[7] https://www.gnu.org/licenses/lgpl.html

Web Server	Tomcat	Tomcat	Tomcat
Licencia	**Commercial Open Source**	**Commercial Open Source**	**Free and Open Source Software**

Tabla 34: Comparativa entre BI Open Source

Hemos destacado la última fila de la comparativa para realizar un breve comentario sobre las licencias. Hemos comentado varias veces que este proyecto tiene una especial sensibilidad al precio por lo que en los casos que hay licencia Open Source comercial hemos analizado con especial interés las limitaciones de la versión gratuita.

En la siguiente tabla compararemos las funcionalidades de cada una de las suites de BI OpenSource. Para la comparativa se han tomado tanto las versiones *Community* como las versiones comerciales (*Enterprise Edition*) de Jasper Soft y de Pentaho. Hemos destacado aquellas características que más se ajustan a nuestro problema

Funcionalidades	SpagoBI	Pentaho	Pentaho Ent. Ed	Jasper	Jasper Ent. Ed
Planificación de Actividades	√	×	√	×	√
Reportes Ad-hoc	×	×	√	×	√
Auditing	√	×	√	√	√
Collaborative BI	√	×	×	×	×
Data Mining	√	√	√	×	×
Dashboard	√	√	√	×	√
Exportación Documentos	√	√	√	√	√
ETL	√	√	√	√	√
Análisis Geo-referenciado	√	√	√	×	√
OLAP	√	√	√	√	√
Query by Example	√	×	×	×	×
Report Validation Workflow	√	×	√	×	×
Reporting	√	√	√	√	√
User Profiling	√	×	√	×	√

Tabla 35: Comparativa de funcionalidad soluciones BI

Como podemos ver no hay gran diferencia entre Pentaho y SpagoBI, sin embargo, la versión gratuita de Pentaho presenta varias limitaciones que pueden ser problemáticas a futuro. Principalmente la carencia de perfilado de usuarios y planificación de ejecuciones.

Para reforzar la toma de la decisión hemos consultado un informe de *Forrester Research* sobre suites de BI Open Source [17]. Como podemos ver en el siguiente gráfico, SpagoBI se encuentra al nivel de las versiones de pago de Pentaho y Jaspersoft según su valoración. La solución que Forrester plantea como la mejor no serviría para nuestro sistema ya que es una herramienta que sólo porporciona funcionalidad de reporting y es demasiado limitada para cumplir con nuestros requisitos.

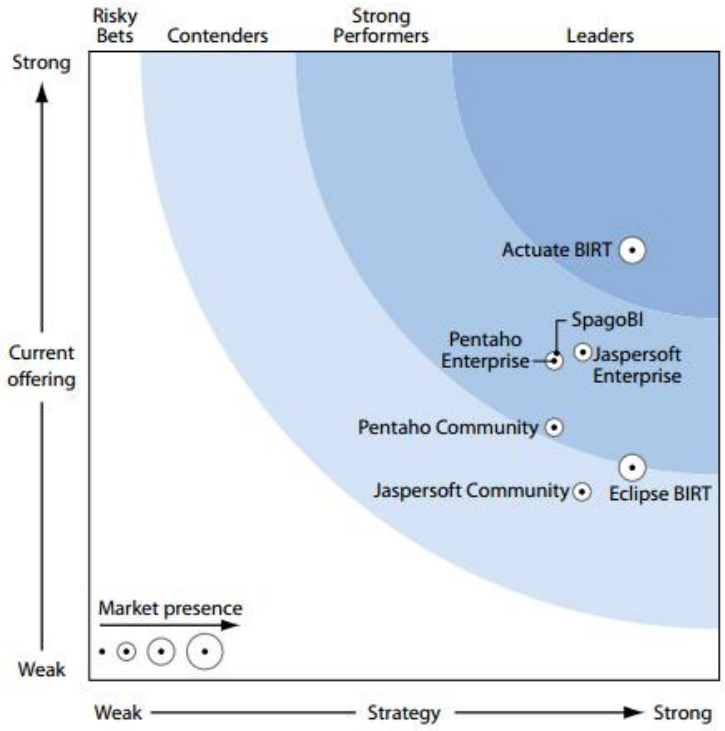


Ilustración 8: Valoración de suites BI Open Source

Valorando todas las consideraciones explicadas con anterioridad la suite BI elegida para nuestra solución sería **SpagoBI.**

4.3.2.2 Módulo de Carga

El módulo de carga es la puerta de entrada de los datos de origen para que estos estén disponibles para la realización de pruebas de auditoría. El módulo de carga debe proporcionar una interfaz a través de la cual se definan:

- Los ficheros que alimentarán la base de datos de auditoría,
- Las transformaciones que se realizarán sobre estos ficheros
- Las tablas de destino de la base de datos

Esta interfaz permitirá gestionar un módulo de carga, cálculo y escritura a la base de datos que se encargará de toda las operaciones lógicas asociadas.

Planteado así el módulo y teniendo como objetivo el uso de componentes *Off the shelf* resulta razonable utilizar una solución ETL (*Extract, Transform and Load)* como módulo de carga. Una solución ETL es un proceso que se encarga de

- **Extraer –** Extrae los datos manual o automáticamente de fuentes heterogéneas.
- **Transformar –** Realiza operaciones lógicas o cruces para almacenar los datos en el formato o estructura apropiado.
- **Cargar (Load) –** Almacenando los datos en una base de datos para su posterior uso.

Pentaho Data Integration

http://community.pentaho.com/projects/data-integration/

Como comentamos en el anterior punto Pentaho es una empresa especializada en soluciones BI *Open Source.* Dentro de su solución BI desarrollaron un módulo ETL conocido como *Kettle* que está disponible como producto independiente bajo la denominación Pentaho Data Integration. Pentaho Data Integration es código libre bajo la licencia *Apache License 2.0* [8]. *Pentaho Data Integration* tiene dos versiones, una *Community* gratuita y una *Enterprise* con mayores funcionalidades pero con coste asociado.

Talend Open Studio

http://www.talend.com/products/talend-open-studio

Talend Open Studio for Data Integration es una herramienta de integración de datos Open Source diseñada para combinar, convertir y actualizar datos de orígenes diversos. La herramienta opera como un diseñador gráfico basado en *Eclipse* que genera programas Java con las transformaciones y cargas de datos estipulados. Los usuarios en Talend diseñan *jobs* basándose en una librería de transformaciones predefinidas o

[8] *http://www.apache.org/licenses/LICENSE-2.0*

transformaciones codificadas por el usuario. *Talend Open Studio* se distribuye bajo la licencia *Apache License v2*[9]. Talend Open Studio también está disponible bajo una licencia comercial denominada *Talend Enterprise Data Integration*.

Característica	Talend Open Studio	Talend Integration Suite	Pentaho Data Integration Community	Pentaho Data Integration Enterprise
Formato de datos de entrada	Ficheros planos, Excel, Bases de datos, XML, PDF y Servicios web.	Ficheros planos, Excel, Bases de datos, XML, PDF y Servicios web.	Ficheros planos, Excel, Bases de datos, XML, PDF y Servicios web.	Ficheros planos, Excel, Bases de datos, XML, PDF y Servicios web.
Bases de datos destino	Oracle, MySQL, PostgreSQL y mayoría de BD comerciales	Oracle, MySQL, PostgreSQL y mayoría de BD comerciales	Oracle, MySQL, PostgreSQL y mayoría de BD comerciales	Oracle, MySQL, PostgreSQL y mayoría de BD comerciales
Planificación de trabajos	Si	Si	No	No
Entorno de desarrollo gráfico	IDE Basado en Eclipse	IDE Basado en Eclipse	IDE basado en SWT	IDE basado en SWT
Diagramas de flujo interactivos	Sí	Sí	Sí	Sí
Detección automatizada de relaciones	Sí	Sí	Sí	Sí
Control de acceso centralizado	No	Sí	No	Sí
Debugger integrado	Sí	Sí	Sí	Sí
Protocolos de comunicación	Web Services, FTP , sFTP, XML files, HTTP, ODBC ficheros planos y MQ Series	Web Services, FTP , sFTP, XML files, HTTP, ODBC ficheros planos y MQ Series	Web Services, FTP , sFTP, XML files, HTTP, ODBC ficheros planos . However no support for Tibco/ MQ Series	Web Services, FTP , sFTP, XML files, HTTP, ODBC ficheros planos
Generación automática de tablas en BD	Sí	Sí	Sí	Sí
Librería de transformaciones predefinidas	Sí	Sí	Sí	Sí

[9] http://www.apache.org/licenses/LICENSE-2.0

Definición de pipelines de transformación	Sí	Sí	Sí	Sí
Permite cruces entre diferentes fuentes para transformaciones	Sí	Sí	Sí	Sí
Licencia	**Open Source**	**Open Source Comercial**	**Open Source**	**Open Source Comercial**

Tabla 36: Comparativa soluciones ETL

Como vemos en la comparativa no hay grandes diferencias en las funcionalidades. Las cuatro soluciones se adaptan bastante bien al objetivo de nuestro módulo. Por razones obvias de coste descartaremos las tres soluciones comerciales (Talend Integration Suite, y Pentaho Enterprise).

De las dos soluciones restantes (Talend Open Studio y Pentaho ETL Community) sólo echaremos en falta la gestión de perfilado en la misma herramienta. Será necesario que cuando se realice la implantación se establezcan medidas mitigadoras a nivel de base de datos o de sistema operativo para garantizar el control de accesos al módulo de carga. Para tomar una decisión entre las dos candidatas restantes tendremos que introducir un nuevo factor que no aparece en la tabla comparativa por su especificidad.

En [18] se incluye un capítulo dedicado a integración de SpagoBI con herramientas ETL y una de las nombradas es Talend. Aunque en la primera versión de la aplicación no se cubra el requisito de la automatización de los procesos de carga, resulta interesante dejar abierta la puerta para, en un futuro, poder automatizar la obtención, transformación y carga de datos combinando Talend y SpagoBI. Por ello el módulo de carga se implementará utilizando **Talend Open Studio.**

4.3.2.3 Base de Datos

El módulo de Base de Datos proporcionará la persistencia de los datos así como un sistema transaccional para hacer consultas. Lógicamente, Como se extrae del análisis de los dos anteriores módulos la elección del Sistema Gestor de Base de Datos no tiene muchos condicionantes ya que ambos módulos tienen compatibilidad con los principales fabricantes del mercado mediante módulos de conexión JDBC/ODBC.

Dado que uno de los requisitos que planteamos fue que los componentes fueran Open Source analizaremos los dos principales gestores de bases de datos Open Source: MySQL y PostgreSQL

PostgreSQL

http://www.postgresql.org/

PostgreSQL, también conocido como *Postgres* es un Sistema gestor de bases de datos relacional cuyo desarrollo pone un énfasis especial en la extensibilidad y el cumplimento de estándares. PostgreSQL implementa el estándar SQL2011 y cumple con la transaccionalidad ACID. Implementa vistas materializadas, *triggers*, *foreign keys*, funciones y procedimientos almacenados.

Es software libre y Open Source liberado bajo la licencia PostgreSQL License, [10].

MySQL

http://www.mysql.com/

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL incluye todos los elementos necesarios para instalar el programa, reparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos.

MySQL ofrece un sistema gestor de base de datos completo incluyendo un motor transaccional de queries basado en SQL, control de privilegios basado en usuarios y roles y distintos sistemas de almacenamiento. Con respecto a los sistemas de almacenamiento de MySQL es importante reseñar que si se quiere tener control de integridad referencial y atomicidad en las transacciones[11]entonces la base de datos se deberá configurar con el motor *InnoDB* en lugar del motor *MyISAM.*

Con respecto a la licencia se dice que es un sistema dual: por un lado se ofrece bajo la GNU GPL v2 para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

[10] http://opensource.org/licenses/postgresql
[11] http://en.wikipedia.org/wiki/ACID

	MySQL	PostgreSQL
SSOO Soportados	FreeBSD Linux OS X Solaris Windows	FreeBSD HP-UX Linux NetBSD OpenBSD OS Xsolaris Unix Windows
Modelo de base de datos	Relacional	Relacional
Esquemas de datos	Sí	Sí
Tipado	Sí	Sí
Indices secundarios	Sí	Sí
SQL	Sí	Sí
APIs y otros métodos de acceso	ADO.NET JDBC ODBC	native C library ADO.NET JDBC ODBC .Net
Lenguajes de programación soportados	Ada C C# C++ D Eiffel Erlang Haskell Java Objective-C OCaml Perl PHP Python Ruby Scheme Tcl	.Net C C++ Java Perl Python Tcl
Server-side scripts	Sí	Funciones definidas por el usuario
Triggers	Sí	Sí
Métodos de particionado	Particionado horizontal en MySQL Cluster	No, se puede conseguir mediante herencia de tablas
Métodos de replicación	Master-master replication Master-slave replication MySQL Cluster	Master-slave replication
Foreign keys	Sí	Sí
Transacciones	ACID	ACID
Concurrencia	Sí	Sí
In-memory capabilities	Sí	no
User concepts	Usuarios con derechos de autorización detallados	Derechos de acceso detalladoe de acuerdo al estándar SQL
Licencia	Open Source	Open Source

Tabla 37: Comparativa SGBD

Analizando esta comparativa no hay grandes diferencias entre una y otra. Las dos bases de datos soportan una funcionalidad parecida y son igualmente compatibles con el resto de módulos propuestos. Es por ello que para seleccionar una u otra optaremos por aquella con la que tenemos más experiencia de desarrollo y administración. En este caso **MySQL es la base de datos elegida** para la implementación de este módulo.

4.3.3 Arquitectura del sistema implantada

En el siguiente diagrama podemos ver la arquitectura que se ha implantado con los módulos seleccionados y la comunicación entre ellos.

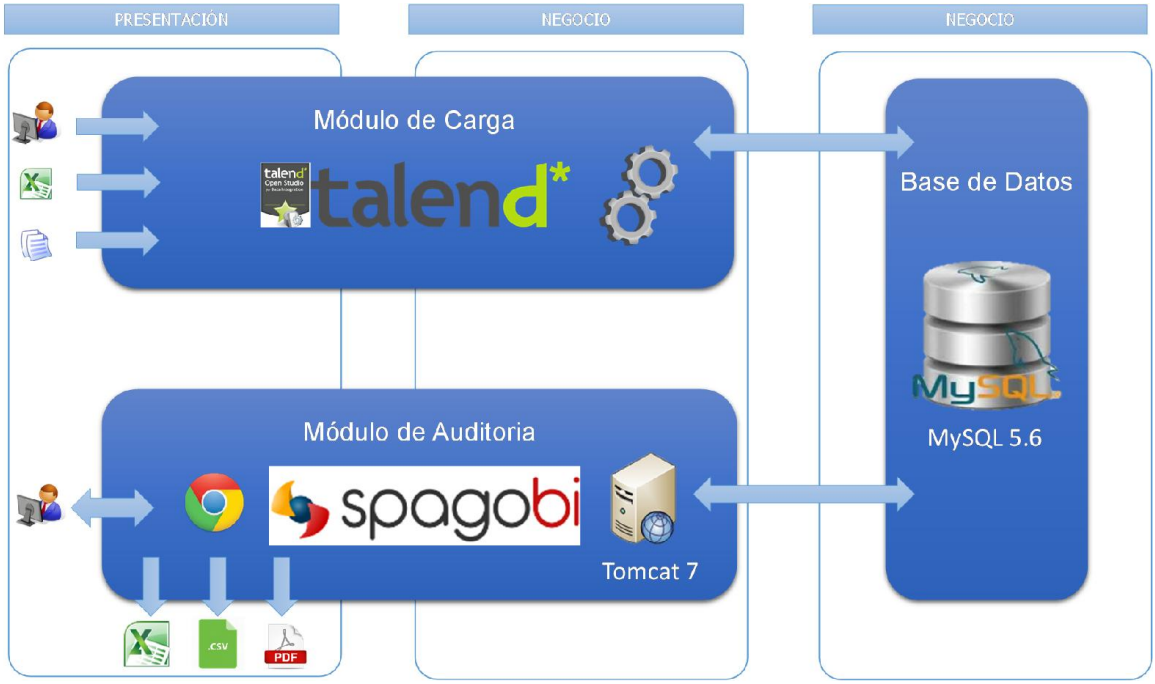


Ilustración 9: Arquitectura del sistema

En la solución implementada no existe comunicación directa entre el módulo de auditoría y el módulo de carga. En versiones futuras se podrá plantear gestionar los jobs de Talend desde SpagoBI incluyendo una nueva comunicación entre módulos que no está representada en este diagrama.

4.3.4 KRIs: Implantación de pruebas de auditoría como KPIs de SpagoBI

En este apartado detallamos la parte del módulo de auditoría que compone el núcleo de las pruebas de auditoría.

Como analizamos anteriormente la herramienta BI *Open Source* que mejor se adapta a nuestros requisitos es *SpagoBI.* Se han analizado las posibilidades avanzadas que proporciona SpagoBI para seleccionar la solución que más se adapte a nuestro problema utilizando el manual [18]

Para el análisis de datos SpagoBI proporciona los siguientes tipos de funcionalidades

- **Reporting** – Genera reportes estructurados parametrizables basados en queries sobre la base de datos. SpagoBI contiene dos motores de reporting BIRT y Jasper Reports.
- **Multidimensional analysis (OLAP)** – El análisis multidimensional permite las consultas jerárquicas de medidas numéricas sobre unas dimensiones predefinidas. El usuario puede monitorizar los datos a diferentes niveles de detalle y desde distintas perspectivas utilizando procesos de *drill-down, drill-across, slice-and-dice y drill-through.*
- **Charts** - El uso de gráficos es lo más habitual para las herramientas de BI ya que permiten simplificar fenómenos complejos en información visual fácil de comprender sin perder semántica.
- **Interactive cockpits** – Los *cockpits* interactivos son visualizaciones compuestas de las otras funcionalidades de SpagoBI. Aunque este tipo de informes son muy utilizados en la industria cuando se implementa una solución BI
- **KPIs** – KPI, como ya hemos comentado, significa *Key Performance Indicator* (Indicador Clave de Rendimiento) y en la implementación de Spag006FBI incluye una serie de métricas, habitualmente derivadas de mediciones simples que permiten presentar un análisis jerárquico de un dominio.
- **Data Mining –** Como comentamos anteriormente, el Data Mining es la extracción de conocimiento implícito previamente desconocido de cantidades masivas de datos mediante el análisis de patrones estadísticos
- **Free Inquiry and driven data-selection –** El Free Inquiry (Consultas libres) permite el acceso fácil a los datos a usuarios sin conocimientos técnicos mediante interfaces gráficas.
- **Ad-hoc reporting –** Como una extensión del Free Inquiry, esta funcionalidad proporciona al usuario no-técnico herramientas para el diseño de sus propios reportes estructurados.
- **Location Intelligence** – Añade un dominio espacial a la información proporcionada por el sistema de BI incluyendo información geográfica al análisis.
- **RT dashboards and consoles –** SpagoBI permite análisis de los datos en tiempo real, de tal manera que se puedan definir unos cuadros de mando o consolas que muestren la información en tiempo real

De estas funcionalidades, a priori las más prometedoras son la funcionalidad de *Reporting* y de *KPIs.* De hecho, para las pruebas de auditoría continua la funcionalidad de KPIs se ajusta de forma óptima ya que es bastante directo convertir las pruebas de auditoría en *KRIs (Key Risk Indicators*[12]*)* que en lugar de medir el desempeño de una métrica midan el riesgo asociado a una prueba de auditoría.

.4.3.4.1 KPIs en SpagoBI

En esta sección vamos analizar el modelo lógico de KPIs que proporciona SpagoBI. Las principales características asociadas a un KPI en SpagoBI son:

- Un KPI se asocia a cálculos complejos mediante lenguaje de script (Javascript o Groovy) o mediante queries sobre una base de datos.
- Un KPI está asociado a un umbral puntual o por rangos para la evaluación de resultados.
- Modelos estructurados en jerarquías con pesos asociados para la generación de KPIs compuestos
- Facilidad de uso, pero con posibilidad de extender a un modelo complejo.
- Un KPI se puede asociar a alarmas cuando un valor cruce los umbrales establecidos

En la siguiente ilustración podemos ver cómo funciona a nivel lógico el modelo de KPIs de SpagoBI. Basándose en los componentes de un modelo KPI (En la columna izquierda) y los datos de entrada almacenados en la base de datos se calculará la valoración de los KPIs que se podrá consultar a través de reportes KPIs definidos. En nuestro dominio, la capa de datos inferior será más simplificada, ya que para realizar pruebas de auditoría no necesitaremos ni un Data Warehouse ni metadatos detallados. Nuestras pruebas correrán sobre una base de datos.

[12] Indicadores Clave de Riesgo

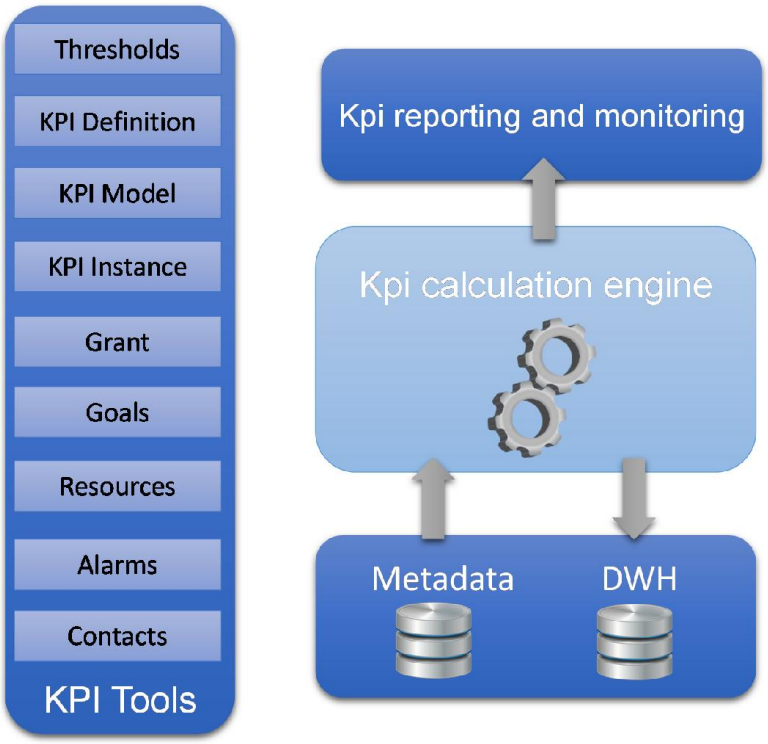


Ilustración 10: Modelo de KPIs en SpagoBI

Un modelo KPI contiene diferentes tipos de metadatos asociados al área que estamos analizando. Algunos de ellos son esenciales para el proceso de definición de los KPIs y deben ser definidos, mientras que otros son opcionales y su uso es discrecional.

Un KPI está asociado a los siguientes elementos en SpagoBI:

- Un nombre, un código único y una descripción
- Un Data Set conteniendo las reglas de cálculo
- Un umbral

El modelo KPI es la estructura jerárquica que organiza los KPIs de acuerdo a una descripción lógica de los procesos que se quieren mediré. Este modelo será la base de los reportes de KPI. Por esta razón la descripción del modelo es una tarea básica para adaptar este tipo de reporting a nuestra aplicación.

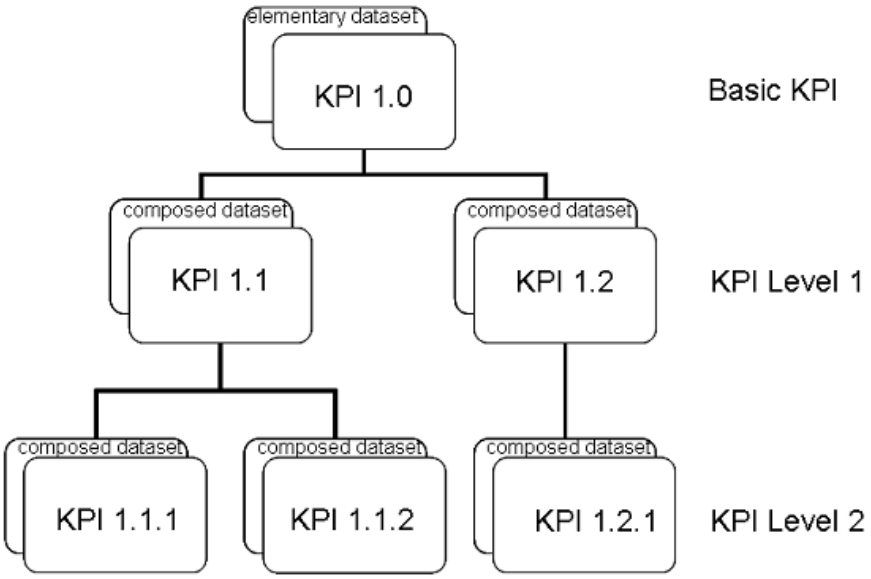


Ilustración 11: Jerarquía de modelo de KPIs

El modelo KPI es una representación lógica pero para que SpagoBI calcule y valore las mediciones de cada KPI este modelo debe ser instanciado en una Instancia KPI. Esta instancia proporciona un recurso ejecutable como representación del modelo. Es posible obtener varias instancias del mismo modelo y cada una de ellas contendrá sus propias mediciones almacenadas.

Dado que los datos de la organización se almacenarán en esta estructura jerárquica de KPIs, creemos interesante realizar un análisis del modelo de datos que utiliza SpagoBI para, al menos, la gestión de KPIs. A nivel interno, el modelo de KPIs de SpagoBI se almacena en una serie de tablas de la base de datos *spagobi*.

El módulo de KPIs almacena los KPIs definidos de forma general en la tabla *SBI_KPI*. Esta tabla contiene un ID único por KPI así como su asociación con el Threshold del indicador. Este KPI se asocia a un KPI MODEL que se almacena en la tabla *SBI_KPI_MODEL* y contiene información jerárquica sobre la relación de los KPIs. Estas dos tablas definen un modelo genérico, que se instancia en las tablas *SBI_KPI_INSTANCE* y *SBI_KPI_MODEL_INST.*

Entre el modelo y la instancia puede haber diferencias, ya que esta es configurable, por lo que la información jerárquica relevante se almacena en cada nodo de la instancia. En el campo *KPI_MODEL_INST_PARENT* hay un enlace al padre del nodo (Siendo este NULL cuando hablamos de una raíz).

Cada Instancia de KPI tiene entre 1 y N KPI VALUES en la tabla SBI_KPI_VALUE cada uno con un rango de fechas de vigencia (Campos BEGIN_DT y END_DT).

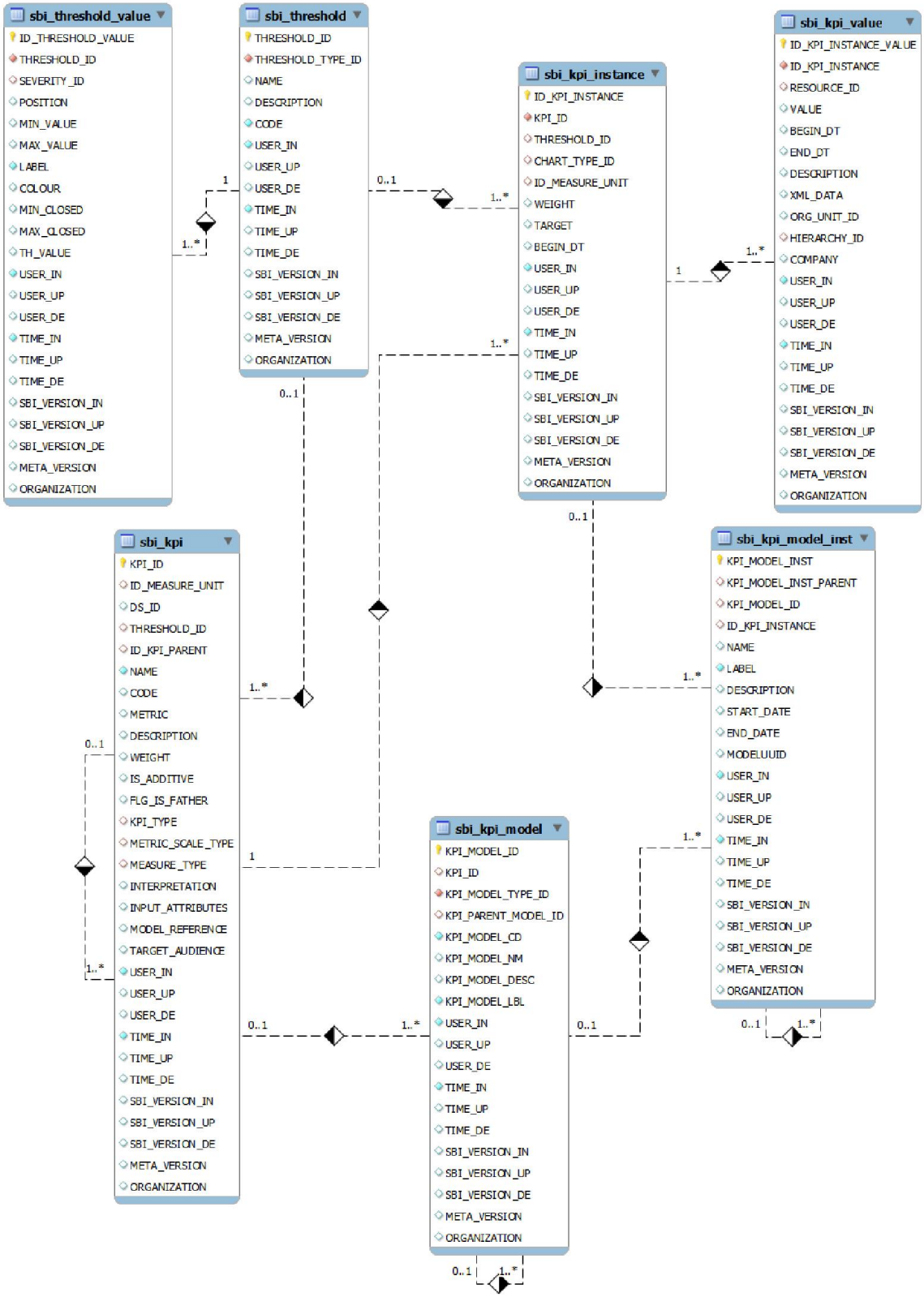


Ilustración 12: Modelo de datos KPI SpagoBI

4.3.5 Conjunto de KRIs propuestos

En este apartado se describirán los KRIs propuestos para la aplicación de auditoría continua en una ficha estandarizada que permita su implementación en SpagoBI. Se han seleccionado una serie de atributos que describen cada uno de los KRIs. Para facilitar su uso como plantilla para el desarrollo de los KRIs en la aplicación se ha presentado un KRI por página.

A continuación se realiza una descripción del significado de cada uno de los atributos utilizados para su descripción:

- Código: Identificación unívoca abreviada del KRI, se construye mediante el código del dominio del KRI seguido de un - y de tres dígitos. Los dominios identificados se corresponden a los dominios de auditoría
- Dominio: Descripción textual del dominio de auditoría al que se adscribe el KRI. Puede ser:
  - Planificación y gestión (PG)
  - Producción (PR)
  - Seguridad (SE)
  - Desarrollo (DE)
  - Aplicaciones (AP)
- Nombre: Identificación extendida del KRI con una descripción del proceso lógico o cálculos a realizar
- Tipo de Archivo Fuente – Descripción lógica del origen de los datos.
- Sistema Fuente – Describe el sistema que genera los datos. Si el origen es un informe manual, se informa de esta característica.
- Umbrales – Código de colores basado en los umbrales de la valoración del resultado del KRI
- Periodicidad – Intervalo en el que se cargan los datos origen del KRI y se recalcula la valoración.

4.3.5.1 Planificación y gestión

KRI			
Código	**PG-001**	**Dominio**	Planificación y Gestión
Nombre	Cumplimiento SLAs		
Descripción	Cálculo del número de SLAs que se cumplen de los proveedores de la organización. Este KRI puede contener información de proveedores de telecomunicaciones, proveedores externos de gestión de la producción, datacenters externos etc.		
Tipo de Archivo Fuente	XLSX - Excel	**Sistema Fuente**	Informes manuales
Umbrales	X < 2 1 <= X > 2 X>= 0	**Periodicidad**	Mensual
Query	`SELECT COUNT(*) AS VALUE FROM` `pg001_sla` `WHERE cumplimiento < sla_threshold;`		

Tabla 38 – KRI PG-001

4.3.5.2 Producción

<table>
<tr><th colspan="5">KRI</th></tr>
<tr><td>Código</td><td colspan="2">PR-001</td><td>Dominio</td><td>Producción</td></tr>
<tr><td>Nombre</td><td colspan="4">% de Backups correctos</td></tr>
<tr><td>Descripción</td><td colspan="4">Calcula el porcentaje de Backups cuyo resultado ha sido el esperado (Que se han ejecutado correctamente) sobre el total de backups planificados en el parque de servidores</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td colspan="2">XLS - Excel</td><td>Sistema Fuente</td><td>Sistema Integrado de Backup</td></tr>
<tr><td>Umbrales</td><td></td><td>X < 95%
95% <= X > 98%
X>= 98%</td><td>Periodicidad</td><td>Semanal</td></tr>
<tr><td>Query</td><td colspan="4"><pre>SELECT
Round(SUM(CASE WHEN BACKUP = 'KO' THEN 1
WHEN BACKUP = 'Undeterminated' THEN 0.5
ELSE 0 END) / COUNT(*), 2)
as value FROM
pr_001_bakup_exec;</pre></td></tr>
</table>

Tabla 39 – KRI PR-001

<table>
<tr><th colspan="4">KRI</th></tr>
<tr><td>Código</td><td>PR-002</td><td>Dominio</td><td>Producción</td></tr>
<tr><td>Nombre</td><td colspan="3">Incidencias críticas en la última semana</td></tr>
<tr><td>Descripción</td><td colspan="3">Informa del número de incidencias críticas (P0-P1-P2) en sistemas de producción registradas en el sistema de gestión de incidencias de la organización Este indicador debe tener umbrales muy estrictos ya que es importante tener constancia de cualquier incidencia crítica en los sistemas de producción</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td>CSV</td><td>Sistema Fuente</td><td>Sistema de Gestión de Incidencias</td></tr>
<tr><td>Umbrales</td><td>X >= 2
X >= 1
X = 0</td><td>Periodicidad</td><td>Semanal</td></tr>
<tr><td>Query</td><td colspan="3"><pre>SELECT COUNT(*) FROM `pr_002_incidences` WHERE `Categoria` IN ('P0', 'P1', 'P2');</pre></td></tr>
</table>

Tabla 40 – KRI PR-002

4.3.5.3 Seguridad

KRI				
Código	**SE-001**		**Dominio**	Seguridad
Nombre	Alertas por virus			
Descripción	Partiendo del reporte mensual de incidencias por virus detectadas en los antivirus corporativos se calcula un % de servidores infectados durante el periodo de cálculo contra el total de servidores con antivirus extraídos de la CMDB de la organización.			
Tipo de Archivo Fuente	XLSX – Excel XLSX - Excel		**Sistema Fuente**	Consola de gestión de antivirus CMDB
Umbrales	X > 5% 5% <= X >= 3% X< 3%		**Periodicidad**	Mensual
Query	`SELECT COUNT(a.equipo)/ (SELECT COUNT(antivirus) FROM xxx_cmdb WHERE antivirus = 'si') FROM` `se_001_virus_alertas` `AS a;`			

Tabla 41 – KRI SE-001

<table>
<tr><th colspan="5">KRI</th></tr>
<tr><td>Código</td><td>SE-002</td><td>Dominio</td><td colspan="2">Seguridad</td></tr>
<tr><td>Nombre</td><td colspan="4">Obsolescencia de SSOO</td></tr>
<tr><td>Descripción</td><td colspan="4">Este KRI parte de dos sistemas, por un lado toma el parque de servidores de la organización de la CMDB y por el otro una tabla compilada por el departamento de sistemas con la obsolescencia de versiones de SSOO basándose en los acuerdos de soporte extendido que tena la organización
En este caso los sistemas operativos pueden tomar un valor ‘Supported’, ‘Unsupported’ u ‘Outgoing’. En el diseño de este indicador se ponderarán los SSOO en estado ‘Outgoing’ (Con soporte extendido, con un valor de 0,5 para el cálculo del porcentaje.</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td>XSLX – Excel
XSLX – Excel</td><td>Sistema Fuente</td><td colspan="2">CMDB
Informe manual</td></tr>
<tr><td>Umbrales</td><td colspan="2">X > 10%
10% <= X >= 5%
X< 5%</td><td>Periodicidad</td><td>Mensual</td></tr>
<tr><td>Query</td><td colspan="4">

```
SELECT
ROUND(SUM(CASE WHEN sup.status = 'unsupported' THEN 1
WHEN sup.status = 'Outgoing' THEN 0.5 ELSE 0 END)
/COUNT(sup.maquina) ,2) AS VALUE
FROM
(SELECT a.*, b.status FROM
XXX_CMDB AS a LEFT JOIN `se002_so_soportados` AS b
ON a.SSOO_version = b.SSOO_version) AS sup;
```

</td></tr>
</table>

Tabla 42 – KRI SE-002

4.3.5.4 Desarrollo

<table>
<tr><th colspan="5">KRI</th></tr>
<tr><td>Código</td><td colspan="2">DE-001</td><td>Dominio</td><td>Desarrollo</td></tr>
<tr><td>Nombre</td><td colspan="4">Proyectos de desarrollo críticos con retraso</td></tr>
<tr><td>Descripción</td><td colspan="4">Partiendo del control de proyectos del departamento de desarrollo se reportará el número de proyectos de desarrollo críticos con un retraso mayor al 10% en la última revisión.</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td colspan="2">XLS – Excel</td><td>Sistema Fuente</td><td>Sistema de control de proyectos</td></tr>
<tr><td>Umbrales</td><td></td><td>X > 5%
5% <= X >= 3%
X< 3%</td><td>Periodicidad</td><td>Mensual</td></tr>
<tr><td>Query</td><td colspan="4">

```
SELECT ROUND(COUNT(*)/
    (SELECT COUNT(*) FROM `de001_informe_desarrollo`
WHERE`Criticidad` = 'Alta') ,2) AS VALUE
FROM `de001_informe_desarrollo` WHERE
`Criticidad` = 'Alta' AND
`Num_horas_actuales`/`Num_horas_estimadas_actuales` >
1.10;
```

</td></tr>
</table>

Tabla 43 – KRI DE-001

<table>
<tr><th colspan="4">KRI</th></tr>
<tr><td>Código</td><td>DE-002</td><td>Dominio</td><td>Desarrollo</td></tr>
<tr><td>Nombre</td><td colspan="3">Proyectos de desarrollo con desviación presupuestaria >15%</td></tr>
<tr><td>Descripción</td><td colspan="3">Partiendo del control de proyectos del departamento de desarrollo se reportará el número de proyectos de desarrollo con una desviación presupuestaria mayor al 15%</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td>XLS - Excel</td><td>Sistema Fuente</td><td>Sistema de control de proyectos</td></tr>
<tr><td>Umbrales</td><td>X > 5%
5% <= X >= 3%
X< 3%</td><td>Periodicidad</td><td>Mensual</td></tr>
<tr><td>Query</td><td colspan="3"><pre>
SELECT ROUND(COUNT(*)/
    (SELECT COUNT(*) FROM `de001_informe_desarrollo`
WHERE`Criticidad` = 'Alta') ,2) AS VALUE
FROM `de001_informe_desarrollo` WHERE
`Criticidad` = 'Alta' AND
`Presupuesto_gastado_actual`/`Presupuesto_Estimado_Actual`
> 1.10;
</pre></td></tr>
</table>

Tabla 44 – KRI DE-002

4.3.5.5 Aplicaciones

<table>
<tr><th colspan="5">KRI</th></tr>
<tr><td>Código</td><td>AP-001</td><td>Dominio</td><td colspan="2">Aplicaciones</td></tr>
<tr><td>Nombre</td><td colspan="4">Cambios en usuarios fuera del cauce habitual</td></tr>
<tr><td>Descripción</td><td colspan="4">Para controlar que el sistema de Altas, Bajas y Modificaciones de usuarios funciona correctamente, el sistema de auditoría continua tendrá una alerta cuando se realicen cambios de emergencia de usuarios en los sistemas de producción. Para ello se filtrarán las RFC (Request for Change) de cuentas de usuario para seleccionar los cambios de emergencia.</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td>XLSX - Excel</td><td>Sistema Fuente</td><td colspan="2">Sistema de gestión de peticiones</td></tr>
<tr><td>Umbrales</td><td>X >=10
10>X>=5
X <5</td><td>Periodicidad</td><td colspan="2">Mensual</td></tr>
<tr><td>Query</td><td colspan="4"><pre>SELECT COUNT(*) AS VALUE FROM `ap001_rfc` WHERE tipo_peticion = 'usuarios' AND categoria = 'Emergencia';</pre></td></tr>
</table>

Tabla 45 – KRI AP-001

<table>
<tr><th colspan="4">KRI</th></tr>
<tr><td>Código</td><td>AP-002</td><td>Dominio</td><td>Aplicaciones</td></tr>
<tr><td>Nombre</td><td colspan="3">Usuarios genéricos</td></tr>
<tr><td>Descripción</td><td colspan="3">Los usuarios genéricos en aplicaciones pueden suponer un problema ya que pueden ser una puerta abierta al acceso no autorizado. Para validar este indicador se cruzará la lista de usuarios de aplicación de gestión de accesos con una lista de usuarios genéricos cargada en el sistema[13]</td></tr>
<tr><td>Tipo de Archivo Fuente</td><td>XLSX – Excel
TXT – Texto plano</td><td>Sistema Fuente</td><td>Sistema de gestión de accesos</td></tr>
<tr><td>Umbrales</td><td>X > 10%
10% <= X >= 5%
X< 5%</td><td>Periodicidad</td><td>Mensual</td></tr>
<tr><td>Query</td><td colspan="3"><pre>SELECT ROUND(COUNT(DISTINCT user_name,system) /
(SELECT COUNT(*) FROM ap002_user_names), 2) AS VALUE
FROM `ap002_user_names` AS a LEFT JOIN
ap002_default_users AS b
ON
a.user_name = b.user
WHERE b.user IS NOT NULL;</pre></td></tr>
</table>

Tabla 46 – KRI AP-002

[13] http://www.sc0rn.com/mi-sc/username-wordlists.zip

4.3.6 Perfilado de usuarios

La auditoría es un campo en el que se trata con datos sensibles. El trabajo de un auditor está regulado por acuerdos de confidencialidad en los que el auditor se compromete a hacer buen uso de la información y no conservarla más allá del transcurso de la auditoría. Nuestra aplicación, sin embargo, actúa como un gran repositorio en el que se almacenará información confidencial tal como usuarios de los sistemas, evaluaciones de seguridad, presupuestos o planes de continuidad de negocio.

Es por ello que un perfilado de usuarios en cada uno de los módulos cobra una especial relevancia, debiendo limitar el acceso a la consulta y modificación de la información en base a unos privilegios definidos para que cada usuario acceda a la mínima información y funcionalidad necesaria para realizar sus tareas.

A continuación describimos el perfilado de usuarios realizado en cada uno de los módulos de la solución implementada.

Módulo de carga de datos

Como analizamos anteriormente la versión libre de *Talend Open* Studio carece de un sistema de perfilado de usuarios integrado. Cualquier usuario que tenga acceso al proyecto de TOS o a los jobs compilados podrá, en principio, realizar cargas en la base de datos utilizando la conexión integrada en Talend.

Por ello, cuando la aplicación se utilice en un entorno de producción es vital la implementación de medidas mitigadoras que impidan la ejecución de los jobs por usuarios no autorizados. En un entorno corporativo esto se podría implementar mediante listas de control de acceso (ACL) en el Active Directory corporativo (Ya que suponemos un entorno Windows). De esta manera sólo aquellos usuarios que determine el propietario de la aplicación podrán ejecutar el módulo de carga de datos.

Módulo de auditoría

El perfilado de usuarios en SpagoBI gestiona por un lado el acceso a la aplicación (Validación de usuarios en la pantalla de login) para a continuación definir qué tareas puede realizar un usuario.

El perfilado de usuarios se realiza mediante asignación de roles a usuarios. Estos roles se asocian a los documentos que componen la funcionalidad de la aplicación .La asociación

entre roles y documentos que gobierna la visibilidad y los permisos de los documentos se denomina en SpagoBI *Behavioral Model*. El *Behavioral model* se basa en 4 conceptos principales:

- **Perfil de usuario**: Definiendo los roles y atributos del usuario
- **Permisos sobre el repositorio**: Definiendo los permisos de accesibilidad del documento
- **Analytical drivers:** Definiendo qué datos se muestran a los usuarios dentro del documento
- **Configuración del entorno de presentación** Definiendo el acceso a los documentos.

En resumen, este modelo presenta respuesta a las siguientes preguntas:

- **Quien** usa la solución (Perfil de usuario)
- **Qué** es visible (Permisos y analytical drivers)
- **Cómo** los usuarios trabajan con sus documentos (Analytical drivers y configuración del entorno de presentación).

Los usuarios en SpagoBI se definen con tres atributos

- Identidades
- Roles
- Perfiles

La identidad de un usuario son los datos que definen al usuario como el username, password y nombre y apellidos.

Los roles definen "qué puede hacer el usuario". Representan una categorización de un grupo de usuario y se asignan para dar permisos sobre documentos o modelos a los usuarios. Todos los roles deben tener un *role type* entre los predefinidos por SpagoBI. Los roles predefinidos son los siguientes.

Tipo de Rol	Descripción	Usuario Standard
ADMIN	Administrador general. Gestiona toda la funcionalidad de SpagoBI	biadmin

MODEL_ADMIN	Administrador de modelos Gestiona el *Behavioral Model* y sus funcionalidades asociadas	bimodel
DEV_ROLE	Desarrollo Crea y modifica documentos BI	bidev
TEST_ROLE	Usuario de Test Pruebas de documentos	bitest
USER	Usuario final Ejecuta documentos sobre los que tenga permiso	biuser

Tabla 47: Tipos de usuario SpagoBI

A cada rol creado por el usuario, además de este se puede asignar una serie de permisos como podemos ver en la siguiente captura de pantalla.

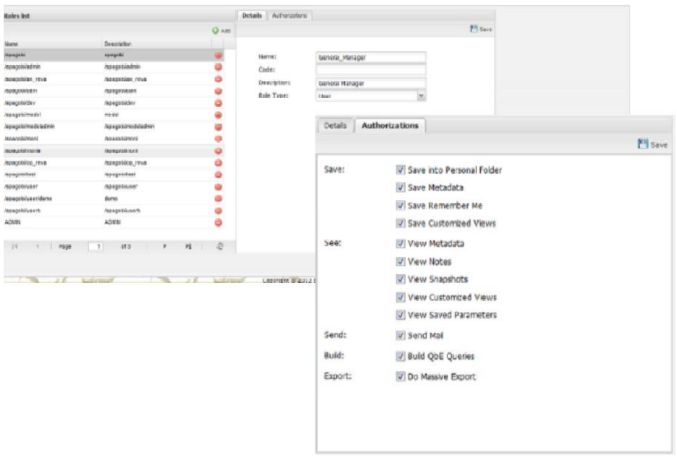


Ilustración 13: Asignación de permisos a un rol SpagoBI

El perfil consiste en una serie de atributos que describen información sobre el usuario (Edad, email…)

La asociación entre roles y documentos, tal y como comentamos antes la asociación entre roles y documentos se realiza a través de los *analytical drivers*. El *analytical driver* modeliza cómo se accede a la información de un documento centrándose en los siguientes factores:

- **Quien** realiza el uso de la información, compilando una serie de roles que tendrán permiso de lectura, modificación o ejecución.

- **Qué** datos se pueden acceder y cómo se presentan a los usuarios. Esta información se almacenan en lo que SpagoBI llama lists of values (LOV).
- **Cómo** verificamos la validez de los valores elegidos. Mediante una serie de *Checks* o controles integrados.

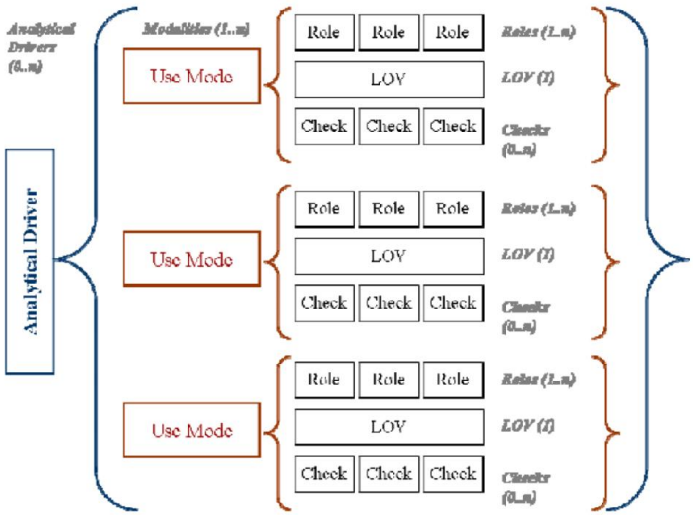


Ilustración 14: Estructura de un Analytical Driver

En lo que a nosotros nos atañe, cada reporte tendrá asociado al menos un analytical driver en el que se seleccionarán los roles que tendrán acceso al documento.

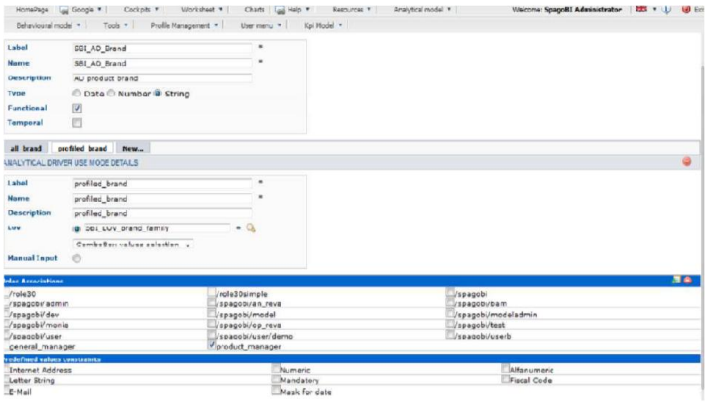


Ilustración 15: Analytical Driver

Para nuestra solución hemos decidido utilizar dos roles de usuario: **User y Admin**. Estos roles diferenciarán entre los usuarios que sólo consultan información de los usuarios que tienen que modificar informes o KRIs en el sistema.

Se deja la puerta abierta a expansiones a futuro tales como nuevos árboles de KRI, una visibilidad de la información basada en la estructura de la organización o simplemente, s o un sistema de permiso más granular. Estas adaptaciones serían sencillas ya que añadiendo nuevos roles podríamos gestionar de forma sencilla el acceso a la información.

Se han creado los siguientes usuarios

Usuario	Tipo básico de usuario	Roles
biadmin	ADMIN	/spagobi/admin /spagobi/user
auditor_jr	ADMIN	/spagobi/admin /spagobi/user
auditor_sr	ADMIN	/spagobi/admin
auditor_mgr	USER	/spagobi/user

Tabla 48: Perfilado de usuarios

Como comentábamos anteriormente, los roles que se asignen se asociarán a *Analytical Drivers* asociados a los documentos de SpagoBI para gestionar el acceso a estos documentos

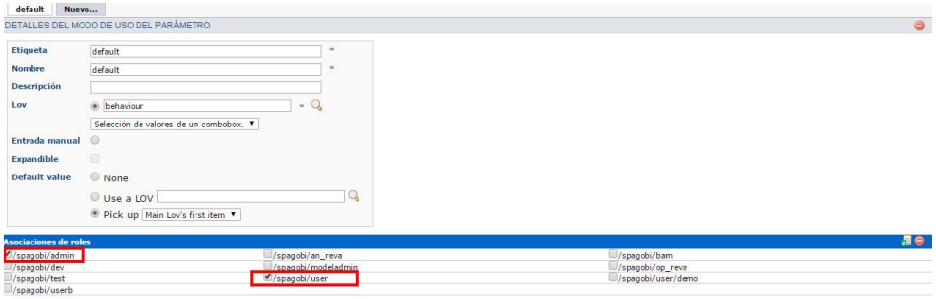


Ilustración 16: Asignación de roles a un Analytical Driver

Como se puede ver la pantalla de inicio de un usuario administrador (Auditor_SR) es distinta a la pantalla de inicio de un usuario de consulta (Manager) representando las distintas funcionalidades a las que tienen acceso.

Ilustración 17: Pantalla inicio usuario Auditor SR

Ilustración 18: Pantalla inicio usuario Auditor_MGR

El nivel de personalización de las opciones de usuario permite configurar, también, la visibilidad de los menús, los documentos accesibles y las operaciones que puede realizar sobre los KPIs (Salvar favoritos, Añadir comentarios…)

Base de datos

Dado que las aplicaciones van a tener acceso de lectura y escritura tanto a la base de datos de pruebas de auditoría (aud) como a la base de datos de spagobi (spagobi) es importante disponer un adecuado perfilado de los usuarios y de una definición granular del acceso.

Se ha definido un usuario por cada tipo de uso en las aplicaciones además de un usuario con permisos de administrador para evitar el uso de cuentas estándar tal como root.

En la siguiente tabla se puede ver el perfilado que hemos realizado para los usuarios de la base de datos. Estos usuarios se utilizarán en las distintas conexiones almacenadas dentro de las aplicaciones.

Usuario	Aud	spagobi	Aplicaciones
uu_fpuskas	√	√	Administración
spagobi_w	X	√	SpagoBI
carga	√	x	Talend
reports	√	x	SpagoBI

Tabla 49: Permisos usuarios BBDD

.4.3.7 Reportes BIRT en SpagoBI

En este apartado analizaremos la funcionalidad de SpagoBI para calcular y presentar reportes de datos estructurados.

SpagoBI hace uso de tecnologías estándar y Open Source en cada uno de sus módulos. Es por ello que SpagoBI proporciona dos tipos de reporting principales *Jasper Reports y BIRT*. No deja de ser un caso similar al que encontramos en la valoración de soluciones ETL, Spago proporciona conexión con las principales soluciones Open Source del mercado independientemente del fabricante. Sin embargo, para limitar la dependencia tecnológica de diversos proveedores vamos a intentar utilizar la solución más estándar, en este caso BIRT.

Los reportes BIRT (Business Intelligence and Reporting Tools) forman parte de un proyecto de la fundación Eclipse para proporcionar un estándar de reporting para aplicaciones Web y Java fácilmente integrable con capacidades para generar reportes en aplicaciones de Business Intelligence.

BIRT tiene dos componentes principales, un diseñador visual de reportes basado en el IDE Eclipse para crear los reportes BIRT y un componente de ejecución para generar reportes que se puede desplegar en cualquier entorno Java. El proyecto BIRT también proporciona un motor de generación de gráficos totalmente integrado con el resto de herramientas. En la siguiente ilustración se puede ver una representación de la arquitectura de BIRT

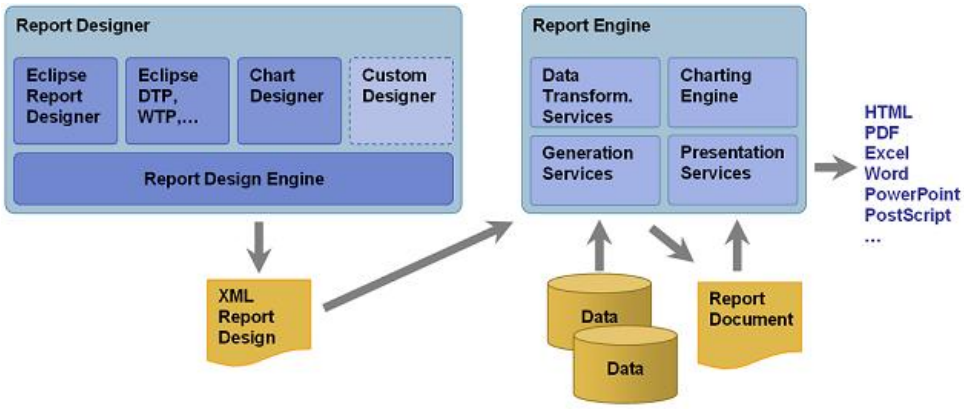


Ilustración 19: Arquitectura BIRT

Para la gestión de salidas en diferentes formatos BIRT proporciona un modelo de componentes basado en lo que denomina *Emitters*. El *Emitter* contiene una lógica para transformar la plantilla XML estándar BIRT y los datos al formato de salida requerido. Los desarrolladores tienen la opción de utilizar los *Emitters* estándar o generar sus propios *Emitters* para obtener los resultados deseados.

En nuestra aplicación definiremos al menos un reporte BIRT por cada KRI propuesto, en el uso de la aplicación por parte de los usuarios los reportes BIRT permitirán a los auditores generar informes sobre las pruebas de auditoría que se pueden realizar en la aplicación.

Los reportes BIRT desarrollados en esta aplicación son los siguientes.

Código	Reporte BIRT	Descripción	Tabla Origen
RP_AP-001	AP-001 Peticiones	Informe detalle de peticiones registradas en el sistema de RFC de la organización	`ap001_rfc`
RP_AP-002	AP-002 Usuarios Genéricos	Cruce entre listado de usuarios genéricos y tabla de usuarios de los sistemas de la organización	`ap002_default_users` `ap002_user_names`
RP_DE-001	DE-001 Proyectos de desarrollo con retraso	Detalle sobre proyectos de desarrollo con cálculo de porcentaje de horas sobre la estimación	`de001_informe_desarrollo`
RP_DE-002	DE-002 Proyectos de desarrollo con desviaciones presupuestarias	Detalle sobre proyectos de desarrollo con cálculo de presupuesto utilizado sobre la estimación	`de001_informe_desarrollo`

RP_PG-001	PG-001 SLA	Detalle de SLAs de la organización con cálculo sobre el cumplimiento de SLAs	`pg001_sla`
RP_PR-001	PR-001 Backups	Reporte detalle de resultado de ejecuciones de backup	`pr_001_bakup_exec`
RP_PR-002	PR-002 Incidencias	Detalle de incidencias registradas en la aplicación de servicio de la organización	`pr_002_incidences`
RP_SE-001	SE-001 Alertas por virus	Detalle de alertas detectadas por los antivirus de la organización	`se_001_virus_alertas`
RP_SE-002	SE-002 Obsolescencia SSOO	Reporte detalle de obsolescencia de SSOO basados en la tabla de estándares de versiones de SSOO de la organización	`XXX CMDB se002_so_soportados`

Tabla 50: Reportes BIRT

Capítulo 5: Ejemplos de uso de la aplicación

En este capítulo se presentarán una serie de ejemplos del uso de las principales funcionalidades que proporciona la aplicación. El capítulo está estructurado de tal manera que se presentan ejemplos de la carga de datos, de la gestión de KRIs y del acceso a la información.

5.1 Módulo de Carga

En el ejemplo sobre el uso del módulo de carga seguiremos paso a paso cómo se define un nuevo trabajo en el módulo de carga y qué sucede al ejecutarlo.

Para el uso del módulo de carga se empezará ejecutando el programa *Talend Open Studio (TOS).*

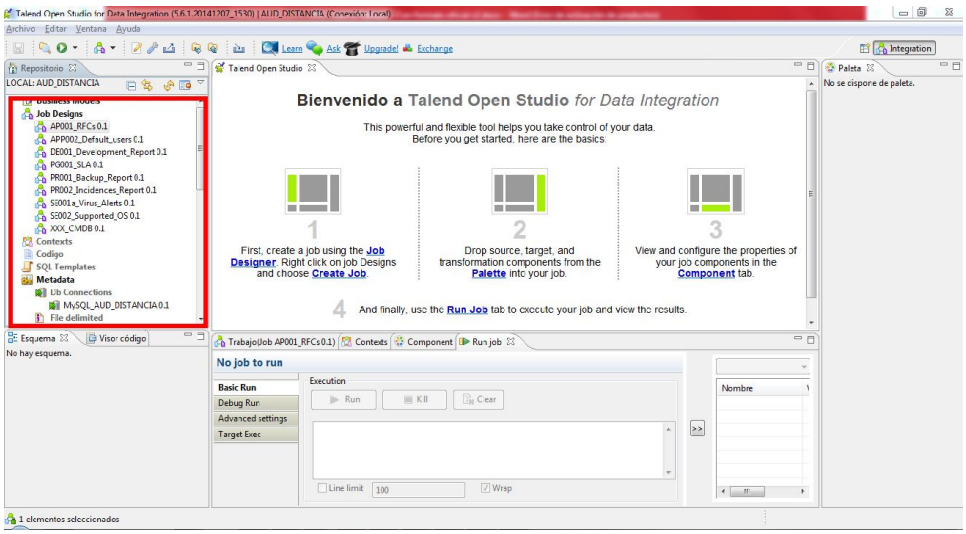


Ilustración 20: Jobs Talend Open Studio

A la derecha de la pantalla de inicio del TOS podemos ver los trabajos que se han definido. Cada trabajo tendrá un código que identifica qué KRI utiliza los datos que carga el trabajo. Se identificará con el código XXX aquellos ficheros que estén asociados a varios KRI.

Si seleccionamos un trabajo se pueden ver los componentes que forman un trabajo desde el fichero de entrada hasta la conexión con la base de datos de salida.

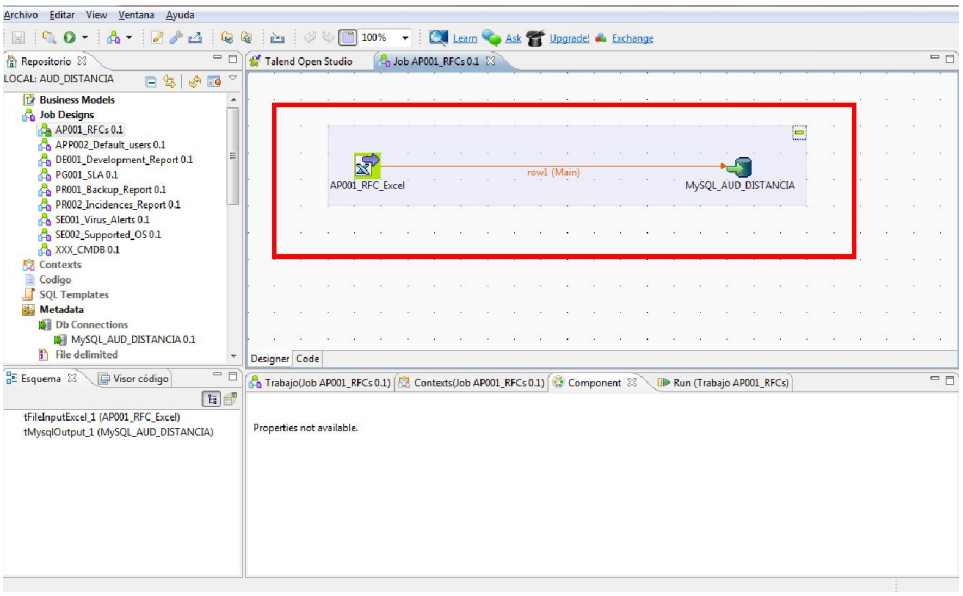


Ilustración 21: Job AP-001

Todos los trabajos diseñados en el proyecto de Talend se pueden exportar como un programa Java standalone mediante la opción *Build Job*.

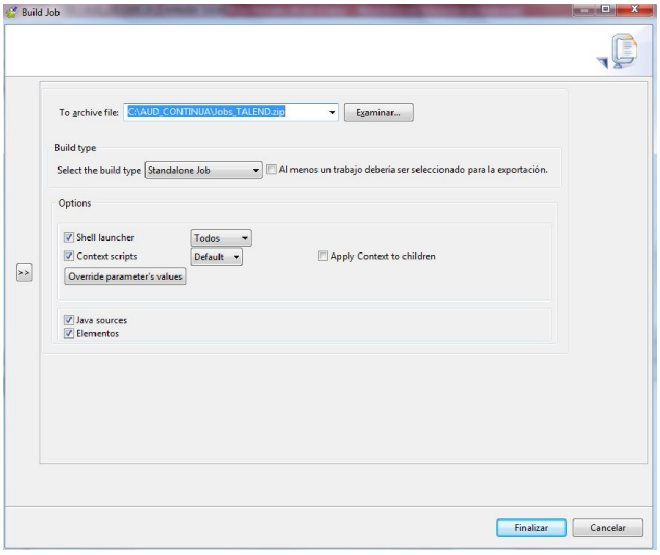


Ilustración 22: Build Job

5.1.1 Creación de un nuevo trabajo en el módulo de carga

Cuando se define un nuevo KRI y se identifican los ficheros fuente de los datos de entrada del KRI se debe crear un nuevo trabajo. En el nuevo trabajo, el primer paso será definir el/los ficheros de entrada, escogiendo el formato elegido. A modo de ejemplo

seguiremos la creación del job AP001 que carga en la BD el fichero de RFCs El primer paso es definir el nombre del fichero de entrada (En este caso un fichero Excel.

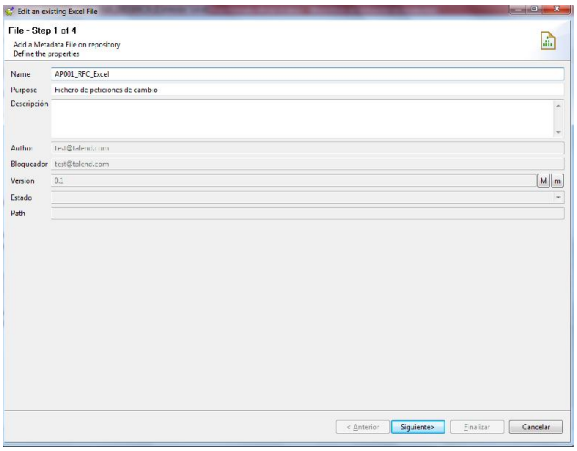


Ilustración 23: Creación de fichero de entrada Excel

A continuación se carga el fichero origen para que Talend pueda procesarlo.

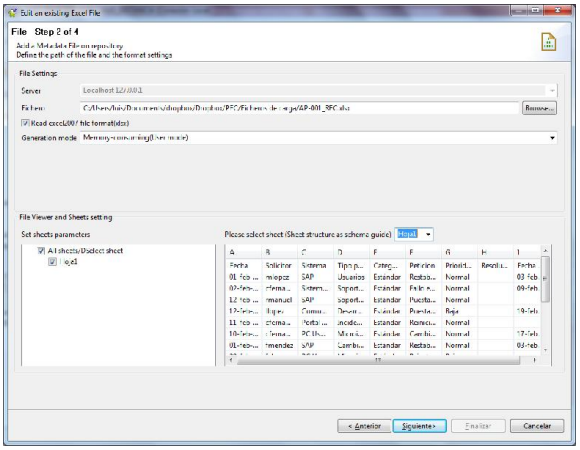


Ilustración 24: Selección de columnas en Talend Open Studio

Y para finalizar Talend reconoce automáticamente los títulos de las columnas como nombre de los campos. En esta pantalla se puede configurar para que ignore una o varias líneas.

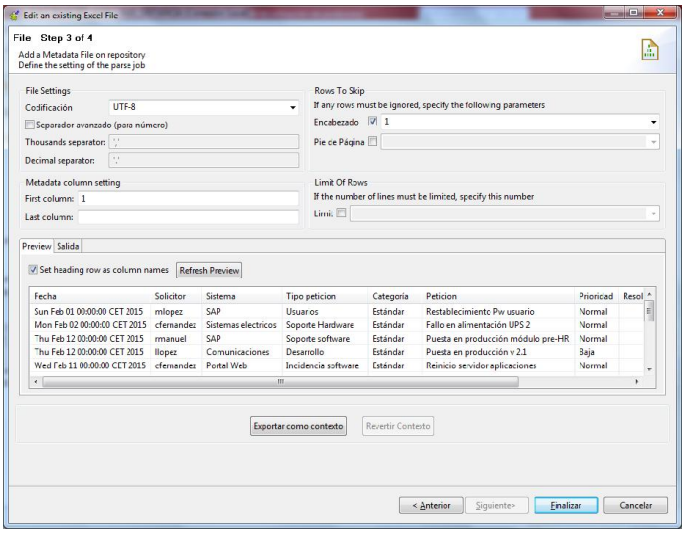


Ilustración 25: Definición de metadatos fichero Excel

Este fichero de carga tendrá asociados unos metadatos que definirán el tipo de datos asociados a cada una de las columnas. Aquí se podrán definir máscaras para transformación de fechas, la posibilidad de hacer *trim* a las columnas para eliminar espacios sobrantes y valores por defecto. Estos metadatos serán la base para la creación de la Base de Datos.

Para la conexión a la Base de Datos en el proyecto Talend definiremos un objeto MySQL.

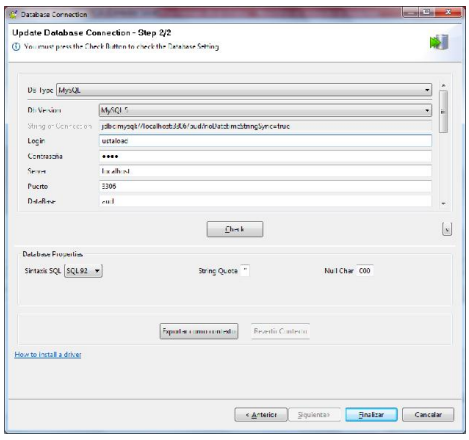


Ilustración 26: Conexión a BD Talend Open Studio

Una vez tenemos los objetos creados, los instanciaremos en el job y los conectaremos mediante una conexión de fila (Botón derecho sobre el fichero de entrada->Fila->Main)

haciendo que los datos fluyan desde el fichero Excel de Entrada a la Base de datos. Para terminar definiremos las propiedades de la base de datos en la pestaña Component

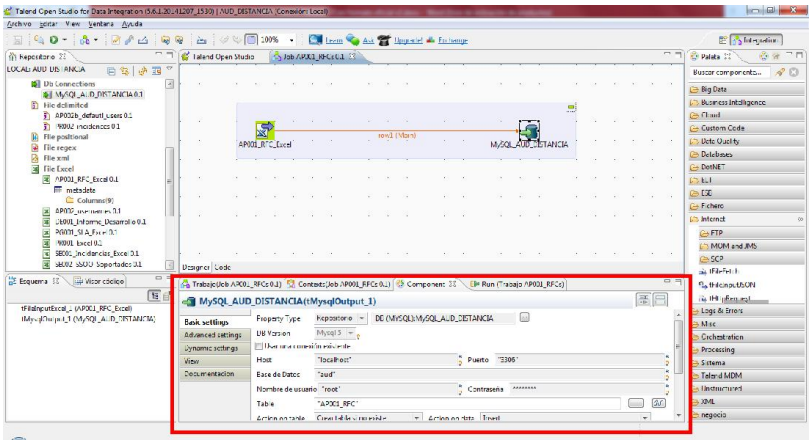


Ilustración 27: Definición comportamiento carga en BD

En esta pestaña definimos dos datos muy importantes. El nombre de la tabla en la que escribiremos y el comportamiento con respecto a la tabla. Se puede configurar para que inserte en la tabla, para que inserte tras realizar un *truncate* a la tabla, que cree la tabla si no existe o que la borre y cree. Nosotros, para mantener un histórico de las mediciones en este caso elegiremos que cree la tabla si no existe. En las configuraciones avanzadas se podrá definir la posibilidad de agrupar los *commit* de la base de datos para realizar inserciones rápidas.

Para terminar vamos a comentar uno de los *jobs* Talend en el que, a modo de ejemplo se ha utilizado un componente que descarga el fichero de Internet.

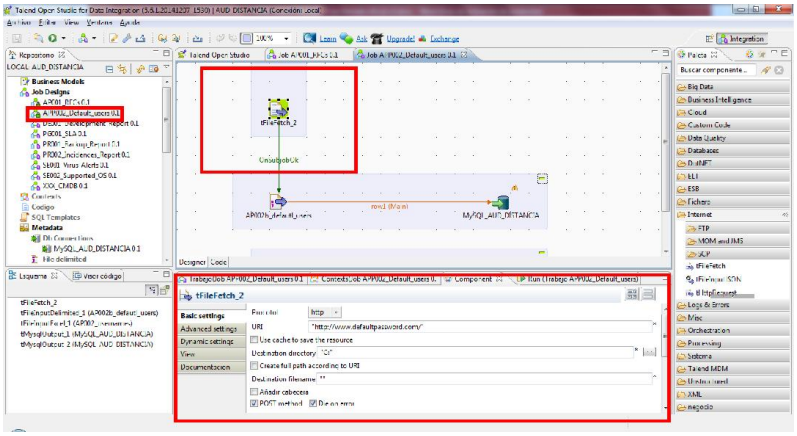


Ilustración 28: Carga de datos desde Internet

Aquí podemos observar cómo hemos creado un componente de tipo tFileFetch que realiza una petición HTTP a la URL que le indiquemos y almacenará la respuesta en un fichero que después trataremos como fichero delimitado por tabuladores. Utilizando este tipo de componentes se podrá automatizar la ejecución de los trabajos haciendo que Talend se conecte a directorios compartidos o servidores ftp para obtener la información.

5.1.2 Ejecución de un trabajo en el módulo de carga.

Para la ejecución de uno de los trabajos, desde la pantalla principal del TOS pulsando en el botón *Run* se despliega una lista con los jobs disponibles. Un clic en uno de los jobs de las lista lanzará su ejecución.

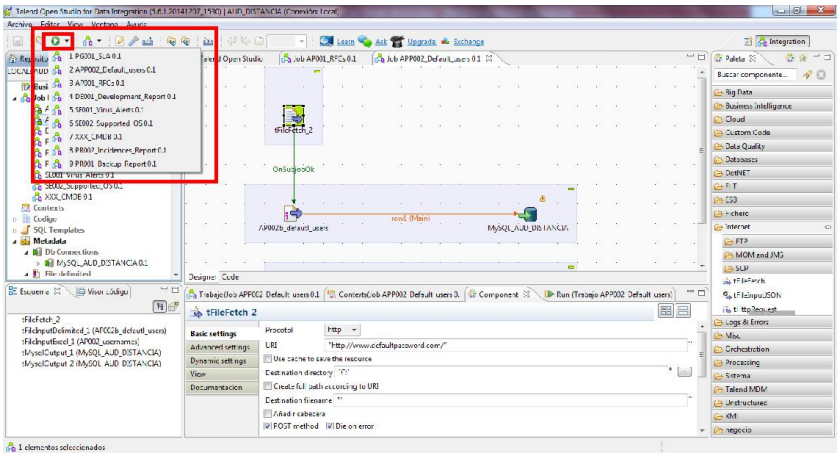


Ilustración 29: Ejecución de job Talend

Si el job no tiene configurada una carga automática de datos será importante comprobar que el fichero que alimenta al componente de entrada es el correcto

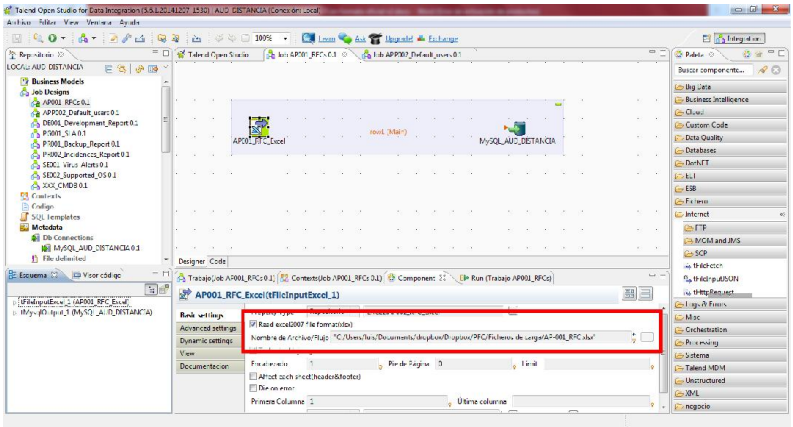
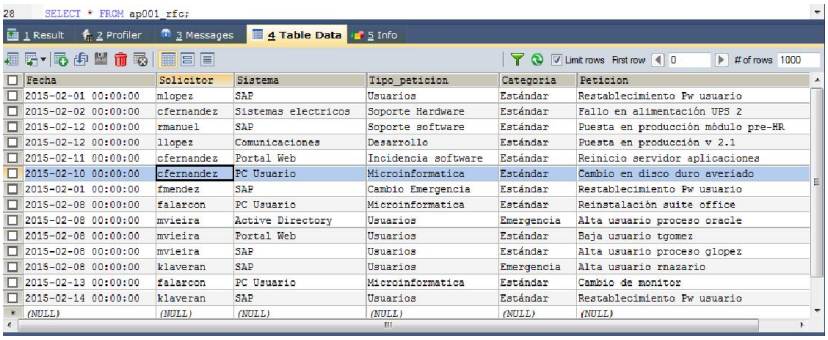


Ilustración 30: Selección de fichero de entrada

Con la ejecución del job Talend informará de lo que ocurre a través de una consola interna. La información que esta consola proporciona es bastante reducida informando solamente de errores importantes y de problemas con los tipos de datos como por ejemplo campos truncados por ser más largos que el tamaño definido en los metadatos.

Para comprobar que la carga fue correcta, conviene realizar una consulta a la base de datos.



Fecha	Solicitor	Sistema	Tipo_peticion	Categoria	Peticion
2015-02-01 00:00:00	mlopez	SAP	Usuarios	Estándar	Restablecimiento Pw usuario
2015-02-02 00:00:00	cfernandez	Sistemas electricos	Soporte Hardware	Estándar	Fallo en alimentación UPS 2
2015-02-12 00:00:00	rmanuel	SAP	Soporte software	Estándar	Puesta en producción módulo pre-HR
2015-02-12 00:00:00	llopez	Comunicaciones	Desarrollo	Estándar	Puesta en producción v 2.1
2015-02-11 00:00:00	cfernandez	Portal Web	Incidencia software	Estándar	Reinicio servidor aplicaciones
2015-02-10 00:00:00	cfernandez	PC Usuario	Microinformatica	Estándar	Cambio en disco duro averiado
2015-02-01 00:00:00	fmendez	SAP	Cambio Emergencia	Estándar	Restablecimiento Pw usuario
2015-02-08 00:00:00	falarcon	PC Usuario	Microinformatica	Estándar	Reinstalación suite office
2015-02-08 00:00:00	mvieira	Active Directory	Usuarios	Emergencia	Alta usuario proceso oracle
2015-02-08 00:00:00	mvieira	Portal Web	Usuarios	Estándar	Baja usuario tgomez
2015-02-08 00:00:00	mvieira	SAP	Usuarios	Estándar	Alta usuario proceso glopez
2015-02-08 00:00:00	klaveran	SAP	Usuarios	Emergencia	Alta usuario rnazario
2015-02-13 00:00:00	falarcon	PC Usuario	Microinformatica	Estándar	Cambio de monitor
2015-02-14 00:00:00	klaveran	SAP	Usuarios	Estándar	Restablecimiento Pw usuario
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Ilustración 31: Comprobación de carga en Base de Datos

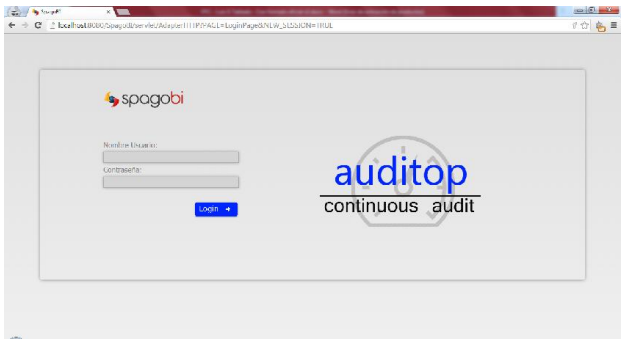
5.2 Módulo de Auditoría

En cuanto al Módulo de Auditoría se han descrito ejemplos acerca de la creación de pruebas de auditoría, la consulta de reportes KRIs y la creación y consulta de informes detallados de datos.

El acceso al módulo de auditoría se realiza a través de un navegador apuntando a la siguiente dirección

http://direccionservidor:8080/SpagoBI

Cuando el usuario accede se le presenta una pantalla de login en la que se le presenta un usuario y un password



Como ya comentamos anteriormente, en función del perfilado del usuario se le presentarán unas opciones u otras. Para los siguientes ejemplos hemos tomado un usuario con permisos de administrador.

En la siguiente captura podemos ver la pantalla de presentación de un usuario al hacer Login. La herramienta permite que cada usuario configure el reporte que él quiera en esta pantalla de tal forma que tenga la información que sea más útil para su trabajo de un vistazo, por ejemplo, se podría configurar para que se presentara el reporte de KRI siempre que se acceda a la aplicación.

Ilustración 32: Pantalla de presentación de la aplicación

5.2.1 KRIs

En este apartado plantearemos un ejemplo acerca de cómo se define un KRI para el cálculo de pruebas de auditoría, desde la definición de la conexión en la base de datos hasta su asignación a una instancia de modelo KRI. También se presentará un ejemplo de acceso de consulta a la aplicación para obtener un reporte de KRI. Es importante informar que en la interfaz de SpagoBI toda referencia a nuestros KRIs se realiza con el término genérico KPI. En este apartado, cuando hablemos de nuestro diseño utilizaremos KRI y cuando hablemos de opciones y menús de SpagoBI utilizaremos el término KPI.

5.2.1.1 Diseño de KRIs

Para implementar una prueba de auditoría en el SpagoBI es necesario que la conexión con la Base de Datos que contiene los datos de origen esté configurada como un Data Source de SpagoBI.

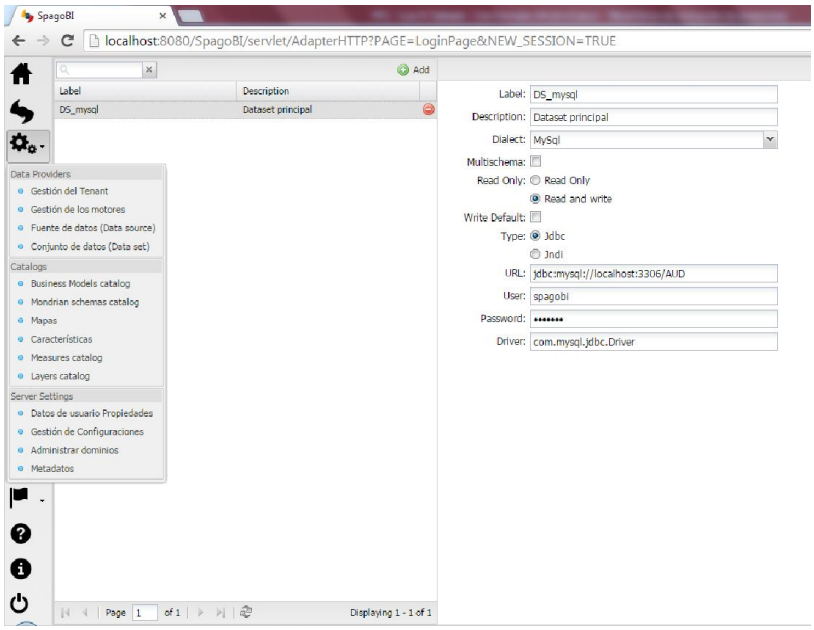


Ilustración 33: Definición de Data Source

Para definir un KRI tenemos que definir sus componentes: Data Set, Threshold y KRI. El Data Set contendrá la prueba en sí ya sea como query MySQL como es nuestro caso o como scritp Groovy o Javascript.

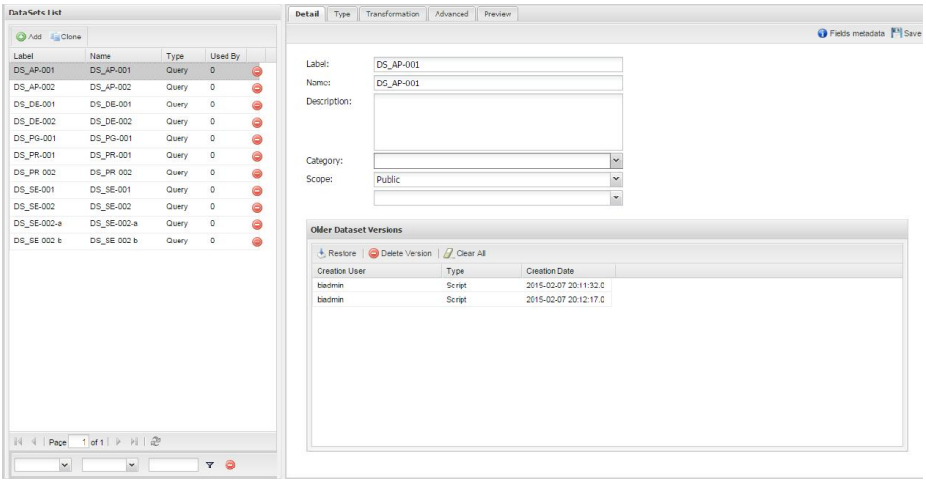


Ilustración 34: Definición de Data Sets

Los Data Sets creados tendrán un código único compuesto por las siglas DS_ y el código del KRI al que van asociados. Este código se introducirá en el campo Label y Name. En la descripción se insertará una descripción textual que ayude a saber lo que hace el cálculo. En la pestaña Type se define la prueba en sí.

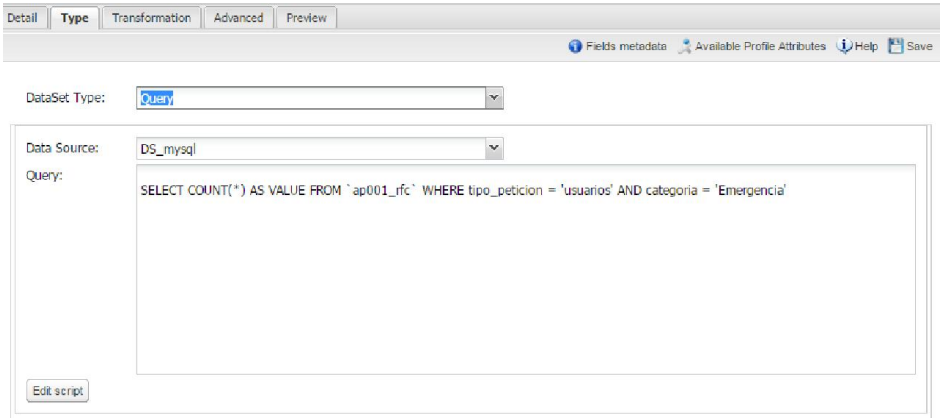


Ilustración 35: Definición de query en Data Set

La query SQL irá asociada al Data Source que definimos anteriormente. La query tiene que devolver un valor numérico en el campo value para que SpagoBI pueda procesarlo. Si se quiere comprobar que la query es correcta, en la pestaña Preview se realiza un cálculo sobre la base de datos.

Basándonos en el análisis de KRI que se presentó anteriormente se definirá un umbral basado en rangos que se asociará al Data Set antes implementado cuando compongamos el KRI.

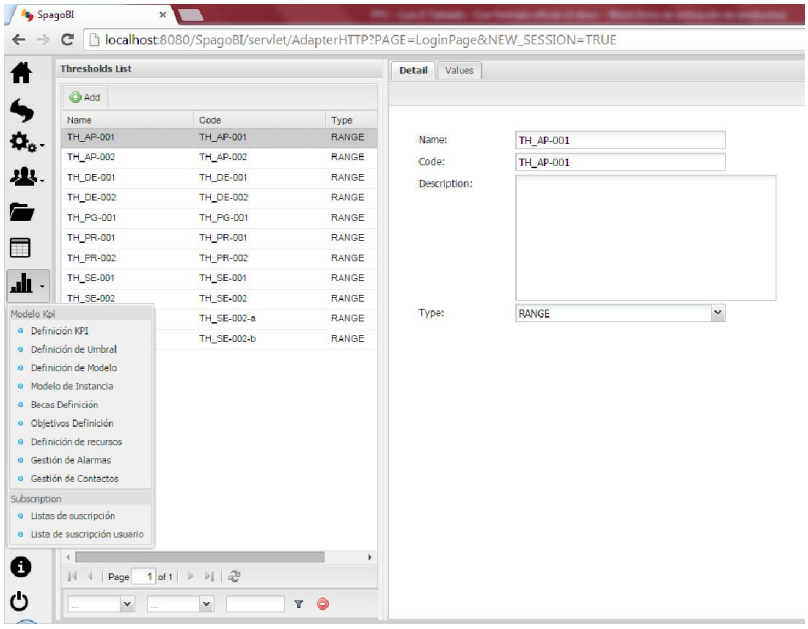


Ilustración 36: Definición de Thresholds

El threshold se definirá, también, con un código único compuesto por el código del KRI precedido de las siglas TH_. En la pestaña values se definen los rangos del threshold.

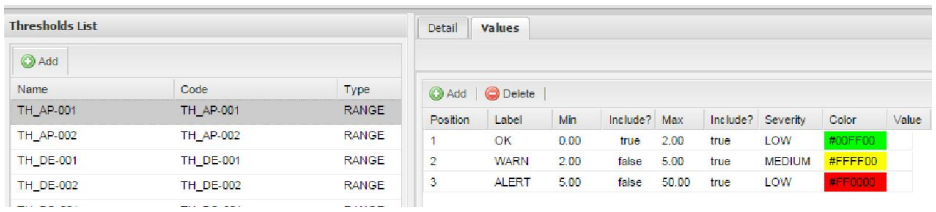


Ilustración 37: Definición de rangos de Thresholds

Los rangos de un threshold están definidos por unos valores máximos y mínimos, una etiqueta, un indicador de severidad estandarizado y una clasificación de colores para ser mostrados en el informe.

Una vez tenemos definido el threshold y el data set ya tenemos los componentes necesarios para definir el KPI. Esta vez el código se compondrá de las siglas KRI_ y el código del KRI que estamos definiendo tal y como se informó en la tabla de descripción de KRIs.

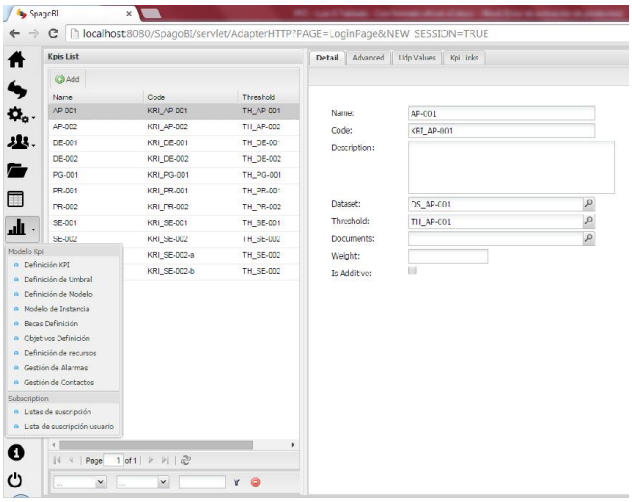


Ilustración 38: Definición de KPI

En esta pestaña se puede ver la posibilidad de asociar, también, un KPI a un documento. Esto permitirá asociar los KPIs a reportes detalle que permitan proporcionar al auditor más información que el cálculo y la valoración.

Una vez creado un KPI por cada KRI definido éstos se agruparán en un modelo jerárquico que definimos en la opción de menú *Model Definition*.

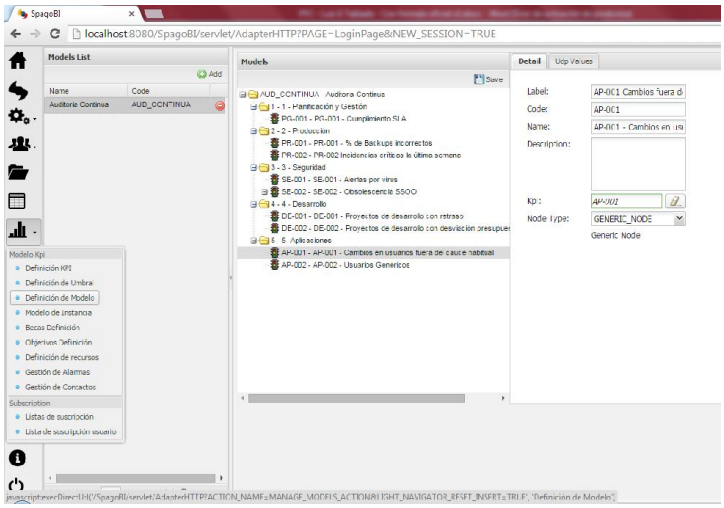


Ilustración 39: Definición del Modelo.

En el modelo que definimos se irán creando los nodos del árbol de dos formas. Por un lado se pueden crear pulsando el botón derecho del ratón y añadiendo nodos vacíos. Por otro lado se pueden arrastrar los KPIs ya creados desde el *sidebar* desplegable de la derecha.

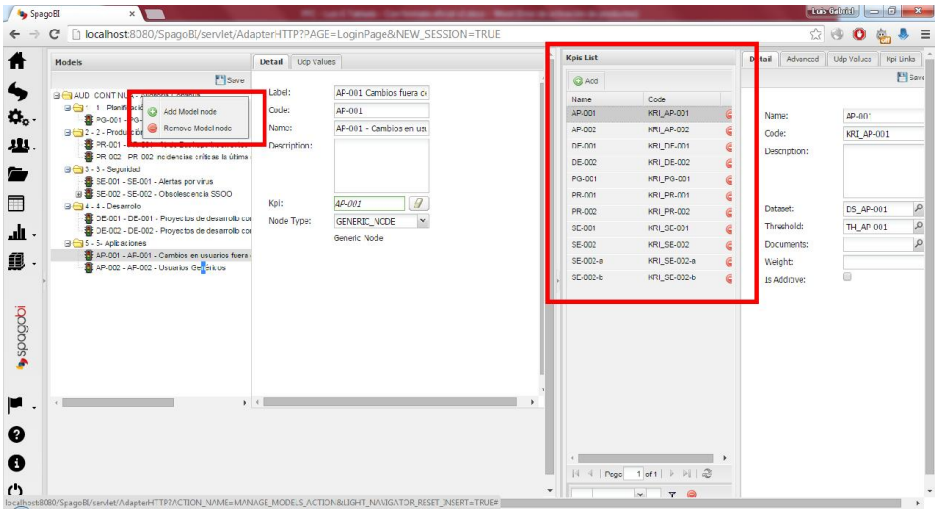


Ilustración 40: Inserción de nodos en el Modelo

Un Modelo de KPI se referencia en una Instancia. Será la Instancia la que esté finalmente asociada a un reporte de KRI y la entidad donde se almacenen los cálculos.Una instancia es, también, un modelo jerárquico en árbol compuesto de nodos que son instanciaciones de los KRIs del modelo. En este caso los nodos del árbol solamente se

crean arrastrando desde el *sidebar* de la derecha que contiene el modelo KPI que definimos con anterioridad.

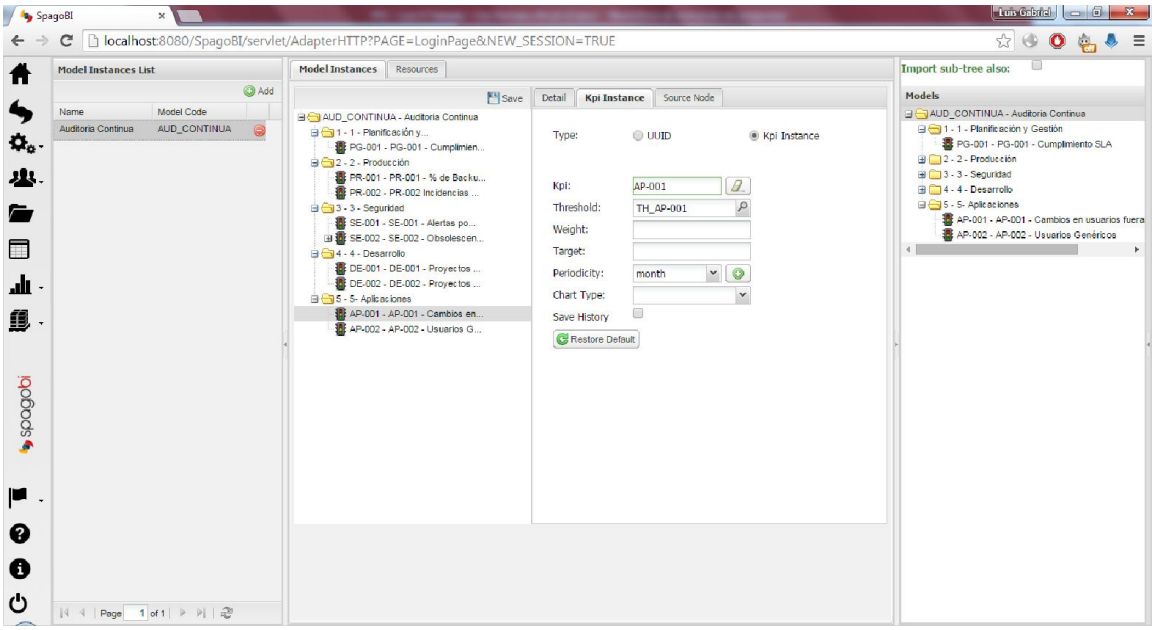


Ilustración 41: Definición de Instancia

Es importante para cada KPR definir la periodicidad del cálculo para que se recalcule tal y como se define en la ficha del KRI.

5.2.1.2 Reporte de KRIs

El acceso a las pruebas de auditoría se realizará a través del informe de KRIs. Este informe presenta todas las pruebas de auditoría en una jerarquía de árbol con sus resultados valorados con un código de colores.

El acceso al reporte de KRIs se realiza por defecto cuando un usuario inicia sesión. También se puede acceder a través del menú de funcionalidades del sidebar izquierdo.

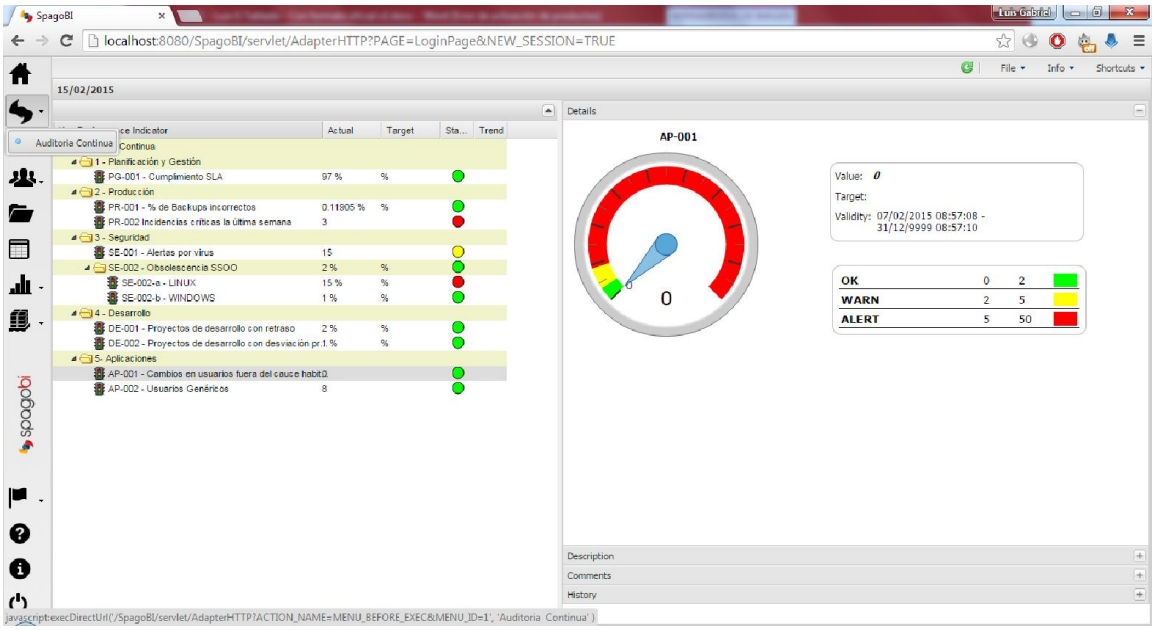


Ilustración 42: Reporte KPI

En el reporte podemos ver el valor de cada uno de los KRIs que hemos implementado y en la partre derecha un velocímetro como representación gráfica de los valores del KRI seleccionado. También proporciona información sobre cuándo se calculó este KRI. En las otras pestañas se puede acceder a una descripción textual, a los comentarios y a un gráfico de tendencia con el historial del valor de este KRI.

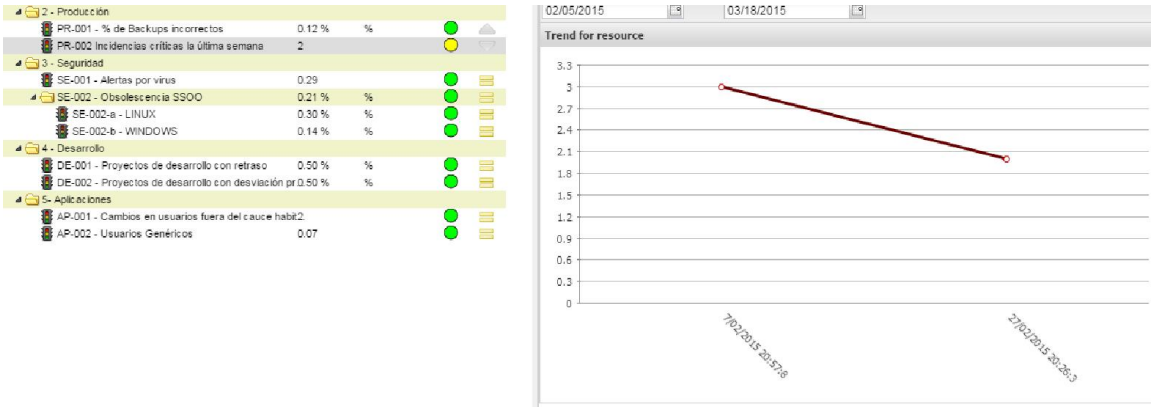


Ilustración 43: Historial de KRIs

5.2.2 Reportes de datos

En este apartado pondremos ejemplos tanto del diseño de reportes de datos mediante el uso de la herramienta *SpagoBI S*tudio, como de la ejecución y asociación de los mismos con KPIs dentro de SpagoBI

5.2.2.1 Diseño de Reportes BIRT

Para diseñar Reportes BIRT tenemos que abrir la aplicación *SpagoBI Studio* y crear un proyecto que esté conectado al servidor SpagoBI en el que se encuentra la aplicación o abrir el proyecto que ya configuramos con la instalación de la solución.

En nuestro proyecto crearemos un reporte BIRT nuevo.

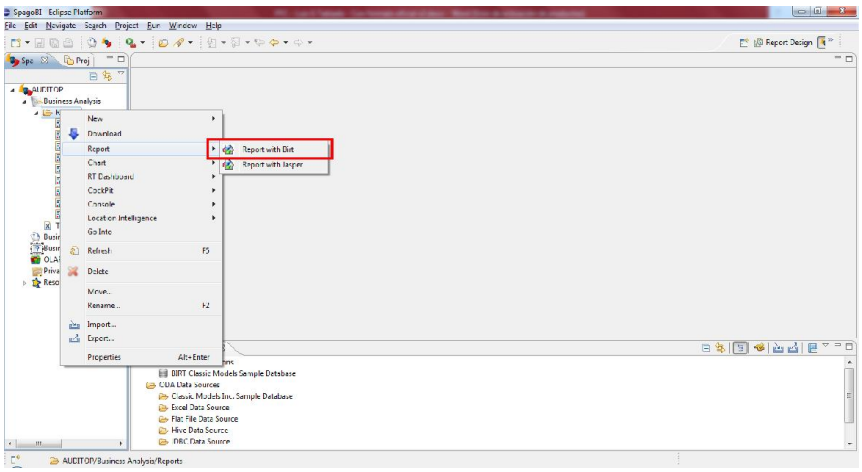


Ilustración 44: Crear nuevo reporte BIRT

Los reportes BIRT siguen el modelo de los reportes que ya hemos visto en SpagoBI. Para crear un reporte BIRT necesitamos definir un Data Source que hará de conexión con la base de datos, un Data Set, que a través de una query SQL devolverá los datos que mostraremos en el reporte y una plantilla para el reporte que definirá cómo mostraremos estos datos.

El primer paso consiste en crear el Data Source como vemos en las siguientes captura de pantalla..

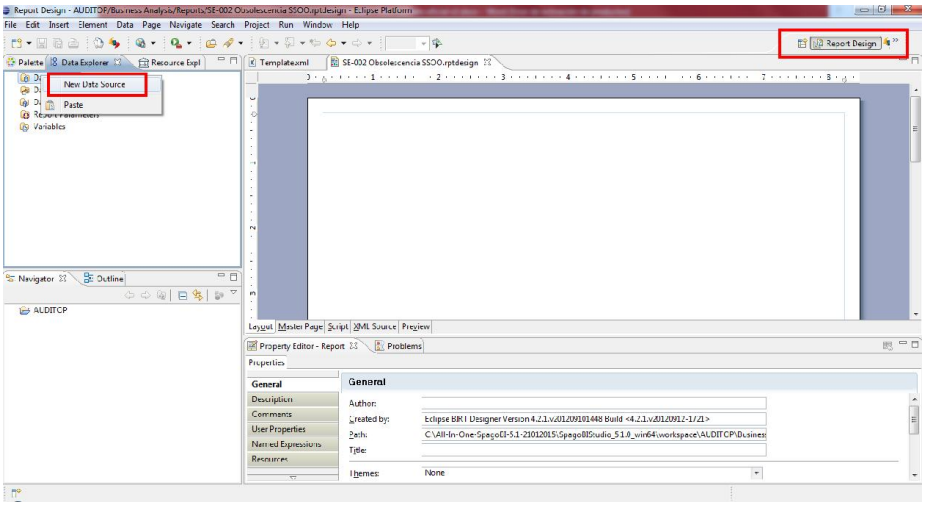


Ilustración 45: Crear Data Source (BIRT)

El Data Source está asociado a una conexión JDBC en la que debemos configurar la dirección del servidor y el usuario con el que nos conectaremos a la Base de Datos.Configuración JDBC

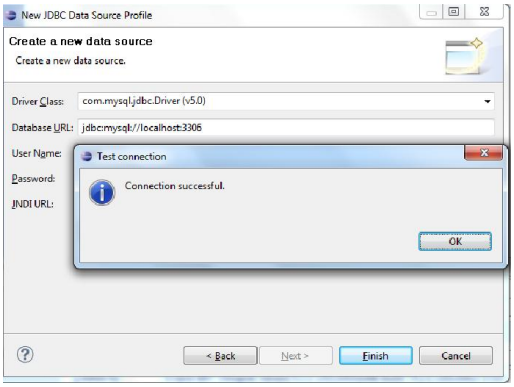


Ilustración 46: Crear Data Source BIRT

En la definición del Data Set indicaremos la query que se ha desarrollado para el reporte. En esta pestaña se podrán utilizar procedimientos almacenados en la base de datos si necesitáramos una lógica más compleja.

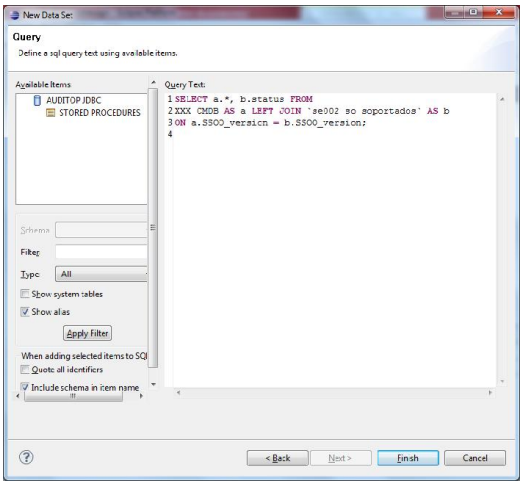


Ilustración 47: Data Set (BIRT)

Una vez introduzcamos la query SpagoBI detectará las columnas de salida y nos permitirá mapearlas para que en el informe aparezcan tal y como queremos.

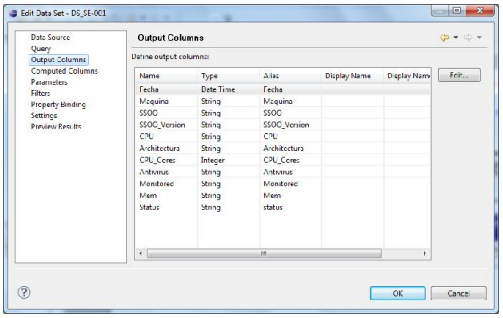


Ilustración 48: Mapeo columnas salida BIRT

Para crear el informe definiremos una plantilla estática con la estética que queramos y para incorporar los datos de la query arrastraremos el Data Set a la plantilla y veremos cómo se crea una tabla que contiene los nombres de las columnas en los encabezados y una fila con los nombres de las columnas entre corchetes, representando que ahí se insertarán los datos. En estas celdas se podrá aplicar un formato individualizado a cada dato en función de cómo queramos mostrar los datos en nuestro reporte.

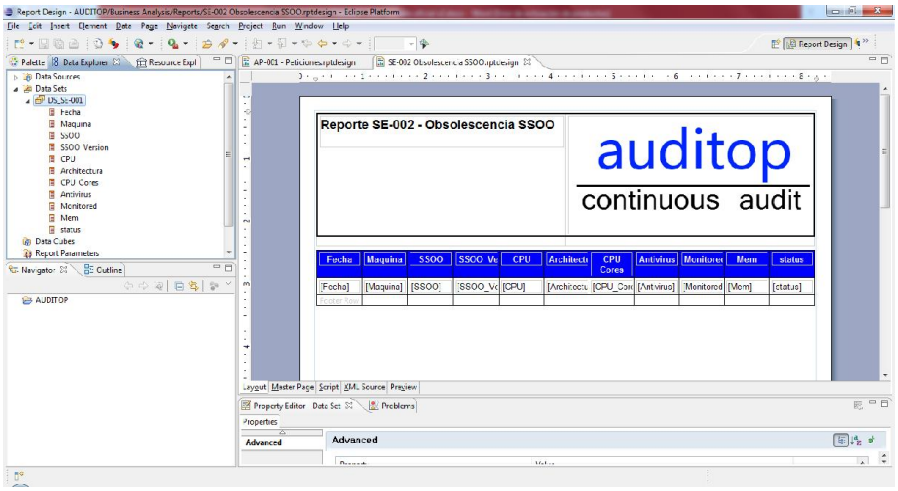


Ilustración 49: Reporte BIR

En la pestaña preview podemos ver como se ejecutaría el reporte BIRT en un navegador.

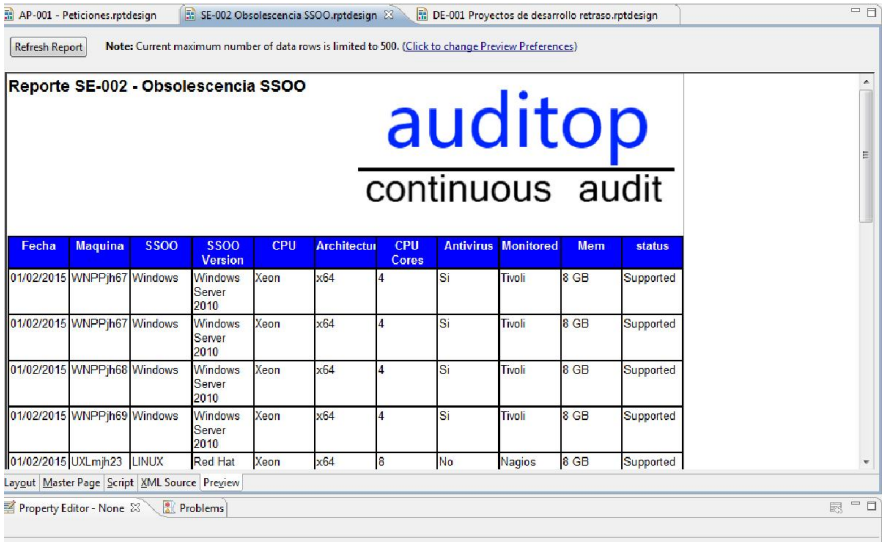


Ilustración 50: Preview de reporte BIRT

Una personalización especialmente interesante es el formato condicional de las celdas de las tablas que nos permitirá que el reporte se adapte dinámicamente en función de la ejecución de la query.

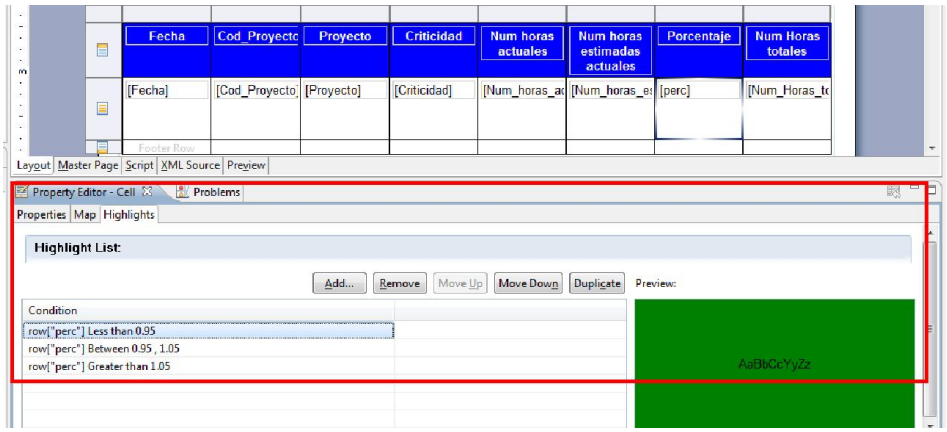


Ilustración 51: Formato condicional reportes BIRT

Una vez el reporte está terminado sólo queda desplegarlo en el servidor para que esté disponible para los usuarios de la aplicación. Este proceso es automático con la opción *Deploy* del SpagoBI Studio.

5.2.2.2 Reporte de Datos

Los reportes de datos estarán disponibles en la interfaz del módulo de auditoría bajo el apartado de reportes. Para ejecutar un reporte simplemente haciendo click sobre el icono hará que la aplicación genere el reporte en tiempo real La aplicación permite a los usuarios marcar como favoritos los reportes para incorporarlos a su carpeta personal y así mantener juntos aquellos reportes que le resulten más útiles.

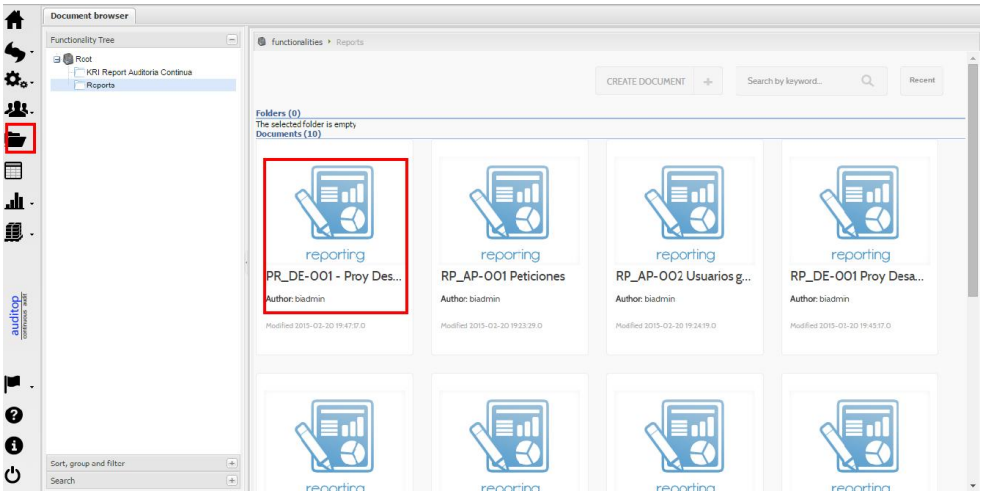
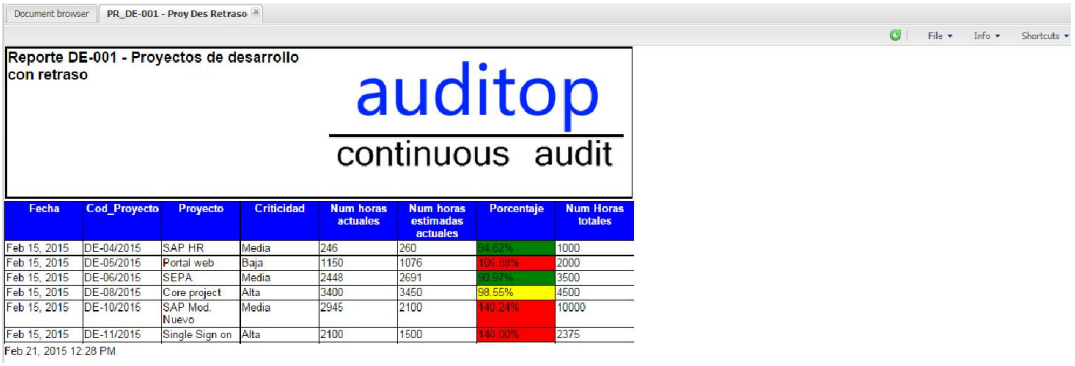


Ilustración 52: Listado de reportes

Como podemos ver en la siguiente captura, el reporte se ha generado como una página HTML.



Reporte DE-001 - Proyectos de desarrollo con retraso

auditop
continuous audit

Fecha	Cod_Proyecto	Proyecto	Criticidad	Num horas actuales	Num horas estimadas actuales	Porcentaje	Num Horas totales
Feb 15, 2015	DE-04/2015	SAP HR	Media	246	260	94.62%	1000
Feb 15, 2015	DE-05/2015	Portal web	Baja	1150	1076	106.88%	2000
Feb 15, 2015	DE-06/2015	SEPA	Media	2448	2691	90.97%	3500
Feb 15, 2015	DE-08/2015	Core project	Alta	3400	3450	98.55%	4500
Feb 15, 2015	DE-10/2015	SAP Mod. Nuevo	Media	2945	2100	140.24%	10000
Feb 15, 2015	DE-11/2015	Single Sign on	Alta	2100	1500	140.00%	2375

Feb 21, 2015 12:28 PM

Ilustración 53: Reporte de datos

La aplicación permite exportar los reportes a los formatos más utilizados (PDF, XLS, CSV, JPG, RTF).

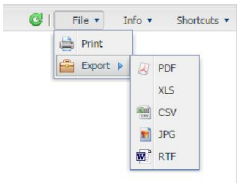
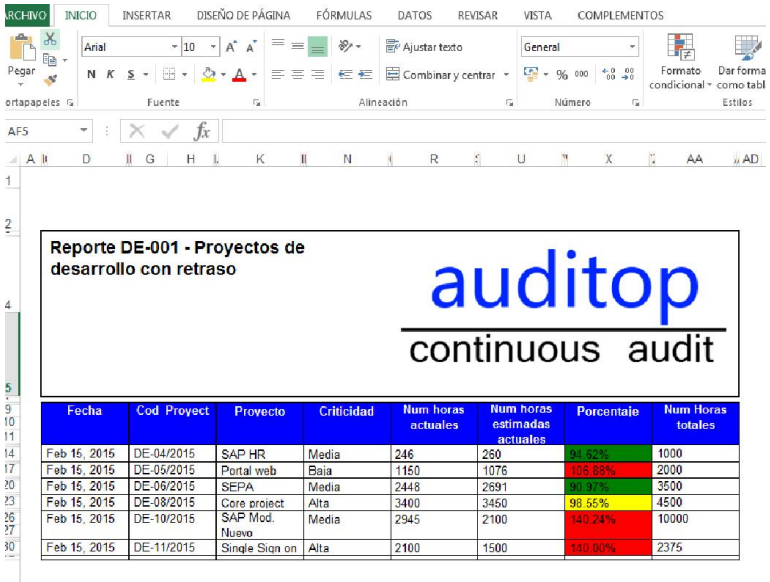


Ilustración 54: Exportación de reportes

Como podemos ver en la siguiente captura, la exportación a Excel mantiene el formato y nos permite disponer de los datos en un formato que podemos tratar fácilmente para realizar nuevas pruebas de auditoría.



Reporte DE-001 - Proyectos de desarrollo con retraso

auditop
continuous audit

Fecha	Cod_Proyect	Proyecto	Criticidad	Num horas actuales	Num horas estimadas actuales	Porcentaje	Num Horas totales
Feb 15, 2015	DE-04/2015	SAP HR	Media	246	260	94.62%	1000
Feb 15, 2015	DE-05/2015	Portal web	Baja	1150	1076	106.88%	2000
Feb 15, 2015	DE-06/2015	SEPA	Media	2448	2691	90.97%	3500
Feb 15, 2015	DE-08/2015	Core project	Alta	3400	3450	98.55%	4500
Feb 15, 2015	DE-10/2015	SAP Mod. Nuevo	Media	2945	2100	140.24%	10000
Feb 15, 2015	DE-11/2015	Single Sign on	Alta	2100	1500	140.00%	2375

Ilustración 55: Reporte en formato Excel

Capítulo 6: Gestión del proyecto

En este capítulo se planteará una planificación del proyecto de diseño e implantación de la solución propuesta en este proyecto de fin de carrera. Para que esta planificación resulte similar a un proyecto real planteamos un equipo de dos personas, un Auditor Senior que actúe como analista y un Técnico que se encargue, principalmente, de las tareas de implantación. Sobra comentar que en el desarrollo de este Proyecto Fin de Carrera ambos roles han sido asumidos por el autor.

6.1 Descripción de las fases del proyecto

Para la gestión del proyecto hemos dividido el proyecto en cinco fases principales que se dividen en trece tareas. Individuales que definimos en la siguiente tabla.

Tareas a Realizar	Código	Descripción
1. Análisis y diseño	**AD**	Análisis y diseño del sistema
1.1 Análisis del problema	AD-1	Análisis del estado del arte de la auditoría continua y de los fundamentos teóricos
1.2 Análisis de requisitos	AD-2	Análisis y generación de casos de uso y requisitos del sistema
1.3 Diseño arquitectura teórica	AD-3	Diseño de módulos abstractos de la aplicación en función de los requisitos estipulados
1.4 Análisis de soluciones COTS	AD-4	Análisis y selección de componentes *Off the Shelf* para implantación de la arquitectura teórica.
1.5 Diseño de arquitectura técnica	AD-5	Diseño final de módulos de la aplicación y de su intercomunicación basado en la arquitectura teórica
2. Implantación	**IP**	Implantación de la solución
2.1 Instalación SW	IP-1	Instalación, Configuración y personalización de los componentes de la aplicación
2.2 Diseño Funcional	IP-2	Diseño funcional de la solución desde el punto de vista de las pruebas de auditoría
2.2.1 Análisis KRIs	IP-2.1	Análisis de un conjunto de KRIs que se podrían implantar en la aplicación. Definiendo qué pruebas de auditoría se van a realizar
2.2.2 Análisis ficheros de entrada	IP-2.2	Análisis de la viabilidad de obtener los ficheros de entrada para los KRIs

		identificando origen y estructura de los ficheros.
2.3 Desarrollo	IP-3	Desarrollo de las pruebas de auditoría diseñadas en la anterior fase
2.3.1 Desarrollo jobs Talend	IP-3.1	Desarrollo de los jobs de carga de datos para los ficheros de entrada.
2.3.2 Desarrollo KRIs	IP-3.2	Desarrollo de las queries de pruebas de auditoría e introducción en el SpagoBI
2.3.3 Desarrollo reports BIRT	IP-3.3	Desarrollo de reportes de datos basados en las pruebas de auditoría
3. Pruebas	**PB**	Testing de la solución integrada
3.1 Pruebas Módulo de carga	PB-1	Pruebas de jobs Talend
3.2 Pruebas Módulo de auditoría	PB-2	Pruebas de cálculo de KRIs y reportes BIRT
4. Generación documentación	**GD**	Memoria del Proyecto y manuales
5. Presentación	**PR**	Preparación de la presentación del proyecto.

Tabla 51: Fases del proyecto

Para la valoración del proyecto las tareas se han definido como no-paralelizables excepto las tareas de pruebas de la aplicación que hemos supuesto se podrían simultanear.

6.2 Planificación

En este apartado se presenta una posible planificación del proyecto. Para realizar la planificación se han supuesto jornadas de 8 horas de dedicación plena. No son estas las condiciones en las que se ha realizado este Proyecto de Fin de Carrera pero en aras de proporcionar un plan de proyecto y presupuesto fideligno con la implantación de la solución diseñada hemos preferido plantearlo así.

En los siguientes subapartados presentaremos una planificación de las fases y las tareas incluyendo una estimación de jornadas por tarea, un diagrama de Gantt de la planificación y un registro del tiempo que se dedicó finalmente a cada una de las tareas.

Tiempo planificado

En la siguiente tabla se presentan las jornadas estimadas por cada una de las tareas a realizar y su sumarización por fase.

Código	Tareas a Realizar	Tiempo planificado (Jornadas)
AD	**1. Análisis y diseño**	13
AD-1	1.1 Análisis del problema	4

AD-2	1.2 Análisis de requisitos	4
AD-3	1.3 Diseño arquitectura teórica	2
AD-4	1.4 Análisis de soluciones COTS	3
AD-5	1.5 Diseño de arquitectura técnica	1
IP	**2. Implantación**	13
IP-1	2.1 Instalación SW	2
IP-2	2.2 Diseño Funcional	7
IP-2.1	2.2.1 Análisis KPIs	5
IP-2.2	2.2.2 Análisis ficheros de entrada	2
IP-3	2.3 Desarrollo	4
IP-3.1	2.3.1 Desarrollo jobs Talend	2
IP-3.2	2.3.2 Desarrollo KRIs	2
IP-3.3	2.3.3 Desarrollo reports BIRT	2
PB	**3. Pruebas**[14]	2
PB-1	3.1 Pruebas Módulo de carga	2
PB-2	3.2 Pruebas Módulo de auditoría	2
GD	**4. Generación documentación**	5
PR	**5. Presentación**	2
	Total	**35**

Tabla 52: Estimación de tiempos por fase del proyecto

En la siguiente página se presentan las tareas planificadas en un diagrama de Gantt

[14] Las tareas de pruebas son paralelizables por diseño ya que los sistemas son independientes

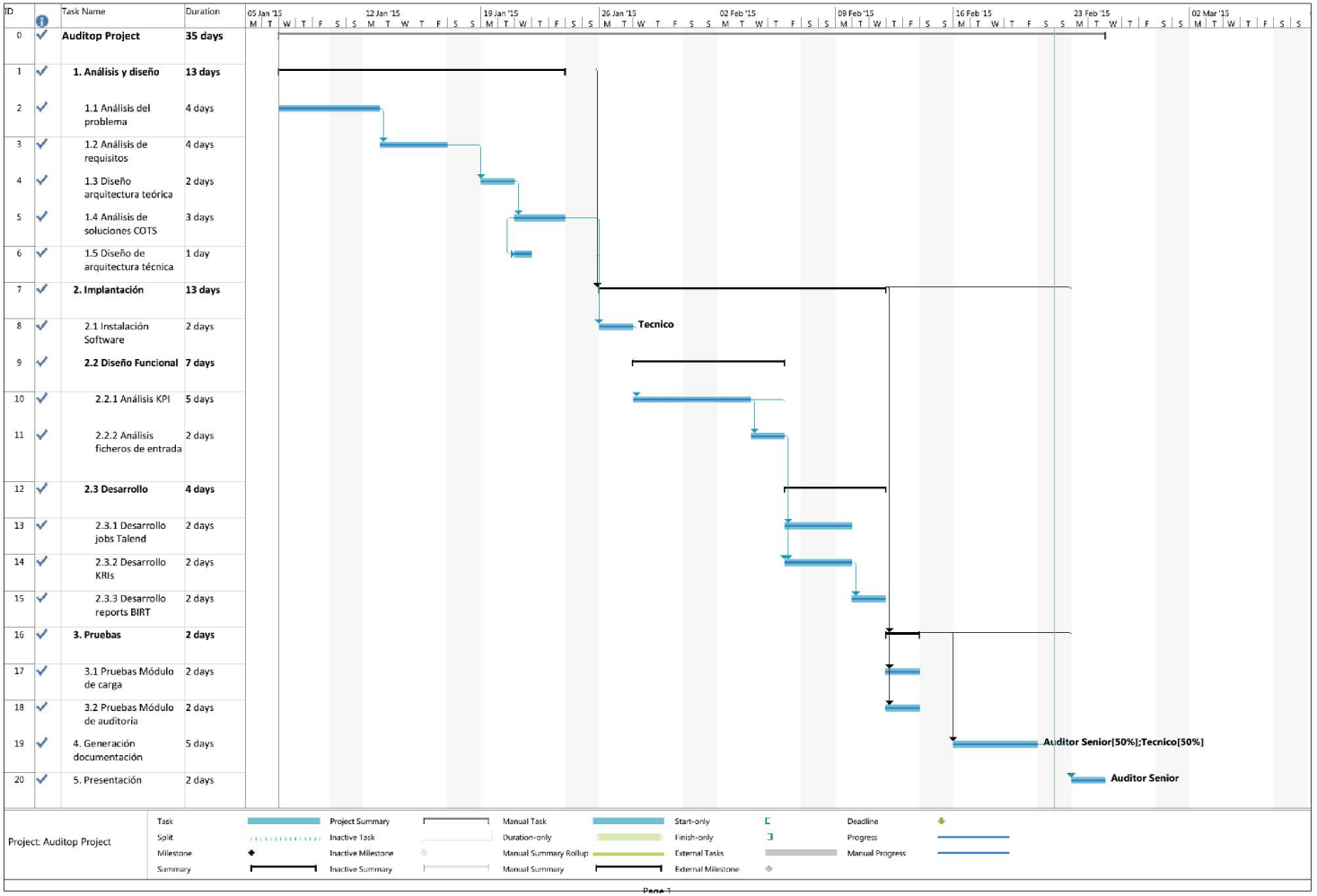


Ilustración 56: Diagrama de Gantt - Estimación de tiempos

Tiempo real

En este apartado proporcionamos un cálculo de los retrasos incurridos desgranándolos por fase. Así, utilizaremos este consumo real de jornadas del proyecto como fuente para el cálculo del coste del proyecto.

Código	Tareas a Realizar	Tiempo Real (Jornadas)	Retraso (Jornadas)
AD	**1. Análisis y diseño**	14	1
AD-1	1.1 Análisis del problema	5	1
AD-2	1.2 Análisis de requisitos	4	-
AD-3	1.3 Diseño arquitectura teórica	2	-
AD-4	1.4 Análisis de soluciones COTS	3	-
AD-5	1.5 Diseño de arquitectura técnica	1	-
IP	**2. Implantación**	14,5	1,5
IP-1	2.1 Instalación SW	3	1
IP-2	2.2 Diseño Funcional	7	-
IP-2.1	2.2.1 Análisis KPIs	5	0,5
IP-2.2	2.2.2 Análisis ficheros de entrada	2	-
IP-3	2.3 Desarrollo	4,5	-0,5
IP-3.1	2.3.1 Desarrollo jobs Talend	1	-1
IP-3.2	2.3.2 Desarrollo KRIs	2	-
IP-3.3	2.3.3 Desarrollo reports BIRT	2,5	0,5
PB	**3. Pruebas**	2[15]	-
PB-1	3.1 Pruebas Módulo de carga	1	-1
PB-2	3.2 Pruebas Módulo de auditoría	2	
GD	**4. Generación documentación**	5	-
PR	**5. Presentación**	2	-
	Total	**37,5**	**2**

Tabla 53: Tiempo incurrido por fase del proyecto

En el siguiente diagrama de Gantt se ve el reparto real de las jornadas. Nótese como el retraso en tareas causa el retraso de aquellas tareas que son dependientes de otras tareas. El retraso fue principalmente en el análisis del problema y en la implantación debido a nuestra falta de experiencia con la auditoría continua y con los módulos seleccionados.

[15] Las tareas de pruebas son paralelizables por diseño ya que los sistemas son independientes

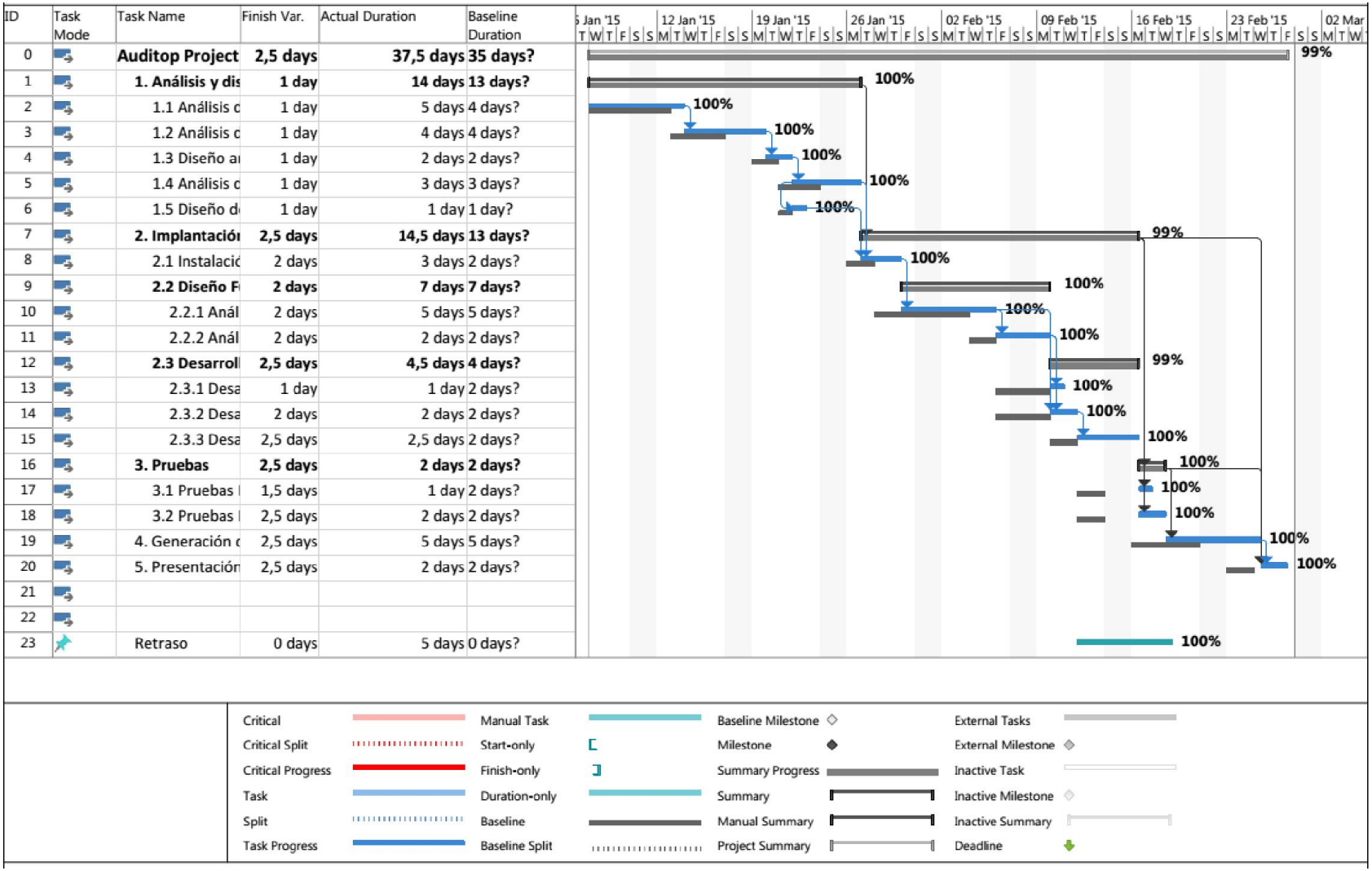


Ilustración 57: Diagrama de Gantt - Tiempos incurridos

6.3 Presupuesto

Para el cálculo del presupuesto debemos tener en cuenta tanto el coste en recursos humanos como el coste material (Hardware, Licencias…)

Recursos Humanos

Para la estimación del coste en Recursos Humanos hemos planteado que el proyecto podría ser llevado a cabo por un equipo de dos personas. Un Auditor Senior con experiencia para las tareas de análisis y tareas generales asociadas a la auditoría y un técnico para las tareas de implantación, instalación y configuración de los módulos de la aplicación.

Las tarifas que aplicamos a cada recurso son las siguientes.

Nombre del recurso	Tarifa
Auditor Senior	30,00 €/hr
Tecnico	20,00 €/hr

Tabla 54: Precio por hora de recursos humanos

En la siguiente tabla se ha plantado un reparto de las horas por tarea pada cada recurso. Tomándose, como dijimos jornadas de 8 horas se ha calculado cuántas horas debe dedicar cada recurso a las distintas tareas. Esta tabla se ha rellenado utilizando las horas incurridas del proyecto, en lugar de las horas planificadas para proporcionar un presupuesto real.

	% Horas Auditor Sr	Horas Auditor Sr	% Horas Tecnico	Horas Tecnico	Horas Tarea	Jornadas Tarea
1. Análisis y diseño	**80%**	**89,6**	**20%**	**22,4**	**112**	**14**
1.1 Análisis del problema	80%	32	20%	8	40	5
1.2 Análisis de requisitos	80%	25,6	20%	6,4	32	4
1.3 Diseño arquitectura teórica	80%	12,8	20%	3,2	16	2
1.4 Análisis de soluciones COTS	80%	12,8	20%	3,2	16	3
1.5 Diseño de arquitectura técnica	80%	6,4	20%	1,6	8	1
2. Implantación	**53%**	**61,4**	**47%**	**54,6**	**116**	**14,5**
2.1 Instalación Software	0%		100%	24	24	3
2.2 Diseño Funcional	100%	56	0%		56	7
2.2.1 Análisis KPI	100%	40	0%		40	5
2.2.2 Análisis ficheros de entrada	100%	16	0%		16	2
2.3 Desarrollo	15%	5,4	85%	30,6	36	4,5
2.3.1 Desarrollo jobs Talend	9%	0,7	91%	7	7,7	1
2.3.2 Desarrollo KRIs	17%	2,2	83%	10,6	12,8	2

2.3.3 Desarrollo reports BIRT	16%	2,5	84%	13	15,5	2,5
3. Pruebas	0%		100%	16	16	2
3.1 Pruebas Módulo de carga	0%			4	4	1
3.2 Pruebas Módulo de auditoría	0%			12	12	2
4. Generación documentación	**50%**	**20**	**50%**	**20**	**40**	**5**
5. Presentación	**100%**	**16**	**0%**		**16**	**2**
TOTALES		**187**		**113**	**300**	

Tabla 55: Asignación de horas por recurso

En la siguiente tabla se agrupa el coste por fase comparando el coste estimado con el coste real de cada una de las fases.

Nonbre Fase	Coste estimado	Coste Real	Variación de coste	Jornadas estimadas	Jornadas reales
1. Análisis y diseño	2.912,00 €	3.136,00 €	224,00 €	13 días	14 días
2. Implantación	2.688,00 €	2.934,00 €	246,00 €	13 días	14,5 días
3. Pruebas	320,00 €	320,00 €	0,00 €	2 días	2 días
4. Generación documentación	1.000,00 €	1.000,00 €	0,00 €	5 días	5 días
5. Presentación	480,00 €	480,00 €	0,00 €	2 días	2 días
TOTALES					

Tabla 56: Coste por fase

En la siguiente tabla se desglosa el coste y tiempo real de trabajo por recurso

Recurso	Coste estimado	Coste Real	Variación de coste	Trabajo Estimado	Trabajo Real
Auditor Senior	5.400,00 €	5.610,00 €	210,00 €	180 hrs	187 hrs
Técnico	2.000,00 €	2.260,00 €	260,00 €	100 hrs	113 hrs

Tabla 57: Coste por recurso

Recursos materiales

De a las consideraciones de diseño tomadas para utilizar software libre y gratuito cuando fuera posible **el coste en licencias de la solución propuesta es cero.**

Para el desarrollo del sistema no hace falta ningún tipo de hardware o software especial. Con un portátil de entorno Windows con 4 GB de RAM se puede correr la solución (Talend, SpagoBI y Base de datos) sin ningún problema. Cuando se pasara a una situación de *Business as Usual* la solución debería ser instalada en un servidor. Una vez más, si la organización en la que se instale no dispone de infraestructura dedicada, el servidor de

aplicaciones Tomcat y el servidor de Base de Datos MySQL pueden compartir la misma máquina.

Entorno	Tipo de sistema	Aplicaciones	CPU	SSOO	RAM
Desarrollo	Portátil	SpagoBI (Tomcat) MySQL SpagoBI Studio TOS	i5 4210u (4 core)	Windows 7 64 bit	4 Gb
Producción	Portátil	SpagoBI Studio TOS	i5 4210u (4 Core)	Windows 7 64 bit	4 Gb
Producción	Servidor	SpagoBI (Tomcat) MySQL	Xeon E5-2407 (4 Core)	Windows Server 2010 64 bit	8 Gb

Coste final

Para el cálculo final del coste hemos considerado utilizar un mismo servidor para desarrollo y producción con entornos virtualizados.

Concepto	Coste
Recursos humanos	5.400,00 €
Licencias	0,00 €
Hardware	1.245,00 €
TOTAL	**6.645,00 €**

Capítulo 7: Conclusiones y trabajos futuros

7.1 Conclusiones generales

En este proyecto hemos realizado una revisión sobre la auditoría de sistemas de información en la que hemos dejado suficientemente claro las tendencias hacia las que se aproxima la profesión. Las organizaciones que no empiecen a utilizar la ingente cantidad de datos automatizados a su disposición para el análisis de sus auditores van a encontrar graves carencias en sus ambientes de control.

En este proyecto hemos realizado una revisión de las líneas de investigación en el campo de la auditoría continua encontrando que, si bien no hay mucha literatura ni proyectos que enfoquen la auditoría de sistemas sí que es un campo en el que se pone gran interés para la auditoría contable.

Hemos encontrado que resulta difícil separar la tarea de auditoría de la monitorización continua, sin embargo creemos que cada campo tiene sus aplicaciones y que la mayor diferencia entre ambas estriba en la introducción del criterio auditor y la existencia de recomendaciones de mejora e informes en la auditoría. Nuestra aplicación ha intentado proporcionar un soporte para que el criterio auditor se pueda incorporar en la figura de las pruebas de auditoría (KRIs) y en la selección de umbrales.

Se tomó la decisión de utilizar herramientas Off the Shelf y Open Source aunque no sea muy común en el marco de las auditoras internas o externas y, consideramos, que ha sido una decisión acertada, tanto como prueba de concepto, como en forma de base sobre la que construir una solución adaptable a cualquier organización.

Aunque no sea una herramienta que se utilice en grandes proyectos (Hemos visto que en grandes organizaciones sigue siendo Pentaho el líder en el marco de BI Open Source) hemos encontrado que la herramienta SpagoBI es muy potente y nos ha proporcionado la funcionalidad que necesitamos. La comunidad detrás de SpagoBI ha resultado de especial ayuda con la información presente en Wikis y Foros.

También nos ha resultado interesante el uso de Talend como solución ETL. Sin embargo debemos destacar que en una implementación a gran escala sería necesario adquirir la licencia para utilizar Talend en un servidor en lugar de como un programa que corre en un PC de usuario.

Consideramos que la solución propuesta es totalmente funcional y serviría para proporcionar auditoría a distancia para una organización de tamaño medio-alto. Si planteáramos este mismo diseño sobre una organización real, con implicación del management para poder tener acceso a los datos y para poder diseñar un conjunto de KRIs suficientemente completo sería razonablemente sencillo tener una herramienta basada en este proyecto que resultara útil para el departamento de auditoría interna.

Si volvemos la vista atrás, en el punto 1.3 de este documento se planteaban tres objetivos fundamentales de este proyecto:

- Analizar la auditoría de sistemas
- Analizar la auditoría continua
- Diseñar e implementar una solución de auditoría continua

Como podemos ver en esta memoria los tres objetivos se han cumplido con creces. Especialmente es reseñable destacar que la solución diseñada e implementada en este proyecto serviría como solución de coste moderado para una organización que quisiera implementar la auditoría continua por necesidades regulatorias o simple interés en mejorar su marco de control.

7.2 Trabajos futuros

Se plantean cuatro líneas principales con las que se podría extender la aplicación para que resultara aún más útil para que una organización cubra su función de auditoría. Estas líneas han sido investigadas dentro del estudio de los módulos de la aplicación y son totalmente realizables. Solamente una falta de tiempo o la imposibilidad de tener acceso a sistemas reales de producción ha impedido que terminen en la versión presentada en este documento. Las líneas de desarrollo futuro son:

Vistas configurables por organización / país

En la solución propuesta hemos planteado una organización sencilla con una serie de KRIs de ejemplo. En nuestro ejemplo un solo árbol de KRIs bastaba para mostrar la información, pero supongamos que la organización en la que se instala la solución fuera una organización con divisiones funcionales o con diversidad geográfica.

Asumiendo una organización con diversidad geográfica. Se puede dar el caso que la aplicación sea accesible por auditores que tengan responsabilidades sobre parte de los

KRI. En este caso en SpagoBI se definiría un modelo que incluyera en el mismo árbol los KRIs de todas las áreas geográficas. Este modelo se instanciaría en un árbol global al que se daría acceso sólo a la parte centralizada.

Para dar acceso a las diferentes áreas geográficas se crearían más instancias que contengan sólo los datos de cada una de las áreas geográficas u organizativas. El acceso a cada una de estas instancias se controlaría de la forma habitual. Mediante asignación de roles al *Analytical Driver* del documento.

Con esta extensión pasaríamos del modelo actual de un árbol de KRI a un árbol KRI global y un número de árboles KRI locales.

Dashboards

Una funcionalidad de SpagoBI que no hemos explotado y sería interesante en versiones futuras es el diseño y utilización de *dashboards* o cuadros de mando. En un *dashboard* se muestra la información más relevante en forma de gráficos o tablas para que los responsables puedan tener todo lo que necesitan para tomar las decisiones adecuadas.

SpagoBI permite incrustar reportes, *widgets* o simplemente el resultado de unas queries en un reporte integrado o en la pantalla de inicio del usuario. Así, con un dashboard, en lugar de encontrarse con una pantalla estática, por ejemplo un manager puede recibir un reporte de aquellos KRIs con estado rojo para ser más eficiente detectando las debilidades de las áreas a auditar.

Conexión automática a sistemas

Aunque hemos explorado en uno de los KRIs descritos la conexión de *Talend Open Studio* con Internet para tomar como origen un fichero descargado este ejemplo dista mucho de ser un proceso puramente automático. Al no disponer de sistemas reales para realizar pruebas no hemos podido implementar un comportamiento que sería ideal en la ejecución de la solución en un entorno de producción.

En un sistema de producción los reportes de los que se nutre la aplicación podrían estar disponibles en un FTP de auditoría. Así las áreas auditadas podrían automatizar la entrega de información y, el trabajo *Talend* se podría ejecutar de forma automática de principio a fin. Encargándose *Talend* de conectarse al FTP, descargar los ficheros, realizar las transformaciones pertinentes y subir los datos a la Base de Datos.

Programación de tareas Talend

Enlazando con el anterior punto de desarrollo futuro aquellos trabajos Talend que fueran puramente automáticos son buenos candidatos para utilizar la integración de programación de trabajos Talend implementada en SpagoBI.

En futuras versiones de la aplicación se intentará integrar el mayor número de trabajos *Talend* dentro del *scheduler* de SpagoBI para que la carga de ficheros se ajuste de forma automática a la periodicidad programada para los KRIs en SpagoBI.

Informe histórico de KRIs

Una limitación importante que hemos encontrado en las opciones que proporciona SpagoBI para consultar la información sobre los KRIs es el acceso a valoraciones históricas de KRI. Como comentamos anteriormente, en el detalle de un KRI se puede acceder a un gráfico histórico de tendencia en el que se pueden ver los distintos valores que ha tomado el KRI a lo largo del tiempo.

SpagoBI no proporciona ninguna manera para generar un informe en el que se obtenga el valor de un KRI a lo largo del tiempo ni su valoración basada en el *threshold.* Es por eso que una línea interesante a futuro sería poder generar una tabla que contuviera esta información. Aprovechando el análisis ya realizado sobre el modelo de datos de KPIs de SpagoBI vamos a esbozar el tipo de informe que se generaría.

Para obtener la relación jerárquica debemos utilizar la tabla SBI_KPI_MODEL_INST. En esta tabla recorreremos el árbol de forma recursiva utilizando el campo KPI_MODEL_INST_PARENT que, como su nombre indica apunta al padre de un nodo dado. Los nodos raíz tendrán este campo valorado como NULL.

Una vez tenemos la representación jerárquica del árbol tendremos que obtener las diferentes valoraciones históricas. Para ello tendremos que cruzar con la tabla SBI_KPI_VALUE para obtener la valoración y la fecha a la que está valorado. SBI_KPI_INSTANCE para obtener qué *threshold* está asociado a ese KPI y SBI_THRESHOLD_VALUE para obtener los rangos y valoración del threshold.

Capítulo 8: Bibliografía

[1] M. d. C. Crespo Rin, El Análisis de Riesgos dentro de una Auditoría, Leganés: uc3m, 2013.

[2] J. Lubbe y F. Snyman, «The advanced measurement approach for banks,» [En línea]. Available: http://www.bis.org/ifc/publ/ifcb33p.pdf.

[3] Bank for International Settlements, «Principles for effective risk data aggregation and risk reporting,» [En línea]. Available: http://www.bis.org/publ/bcbs239.pdf.

[4] ISACA, «ITAF, 3rd Edition,A Professional Practices Framework for IS Audit / Assurance,» [En línea]. Available: http://www.isaca.org/Knowledge-Center/Research/Documents/ITAF-3rd-Edition_fmk_Eng_1014.pdf.

[5] ISACA, «Principios COBIT según la ISACA,» [En línea]. Available: http://www.isaca.org/COBIT/Pages/default.aspx. [Último acceso: 01 02 2015].

[6] ISACA, CISA Review Manual 2014, 2014.

[7] ISACA, «Standards for IS Audit and Assurance,» [En línea]. Available: http://www.isaca.org/Knowledge-Center/ITAF-IS-Assurance-Audit-/IS-Audit-and-Assurance/Pages/Standards-for-IT-Audit-and-Assurance-English.aspx. [Último acceso: 24 01 2015].

[8] A. Kogan, E. E. Sudit y M. A. Vasarhelyi, «Continuous Online Auditing: An Evolution,» 1999. [En línea]. Available: http://raw.rutgers.edu/docs/research/ON-AUD15f-1.doc.

[9] K. Handscombe, «Continuous Auditing from a practical perspective,» [En línea]. Available: http://www.isaca.org/Journal/Past-Issues/2007/Volume-2/Pages/Continuous-Auditing-From-a-Practical-Perspective1.aspx. [Último acceso: 07 02 2015].

[10] G. Brennan, «Coninuous Auditing Comes of Age,» [En línea]. Available: http://www.isaca.org/Journal/Past-Issues/2008/Volume-1/Documents/jpdf0801-continuous-auditing.pdf.

[11] C. E. Brown, J. A. Wong y A. A. Baldwin, «Research Streams in Continuous Audit: A Review and Analysis of the Existing Literature,» [En línea]. Available: http://raw.rutgers.edu/docs/wcars/12wcars/BrWoBa.12CA.pdf.

[12] M. Bovee, A. Kogan, K. Nelson, R. P. Srivastava y M. A. Vasarhelyi, «Financial Reporting and Auditing Agent with Net Knowledge (FRAANK) and eXtensive Reporting Language,» *Journal of Information Systems,* vol. 19, nº 1, pp. 19-41, 2005.

[13] M. A. Vasarhelyi, M. Alles, S. Kuenkaikaewa y J. Littley, «The acceptance and adoption of continuous auditing by internal auditors: A micro analysis,» *International Journal of Accounting Information Systems,* vol. 12, nº 3, pp. 267-281, 2012.

[14] M. A. Vasarhelyi, S. Romero, S. Kuenkaikaew y J. Littley, «Adopting Continuous Auditing/Continuous Monitoring in Internal Audit,» *ISACA Journal,* vol. 3, 2012.

[15] D. Coderre (RCMP), GTAG 3: Continuous Auditing: Implications for Assurance, Monitoring, and Risk Assessment, The Institute of Internal Auditors, 2005.

[16] M. Golfarelli, «Open Source BI Platforms: a Functional and Architectural Comparison,» *International Conference on Data Warehousing and Knowledge Discovery,* vol. 5691, pp. 287-297, 2009.

[17] Forrester Research, «The Forrester Wave™: Open Source Business Intelligence (BI), Q3 2010,» [En línea]. Available: http://www.actuate.com/ger/download/wave_open_source_business_intelligence_bi_q3_2010.pdf.

[18] SpagoBI Competency Center , Business Intelligence with SpagoBI, 2012.

Capítulo 9: Anexos

En este capítulo se describen una serie de anexos que resultan de utilidad para el despliegue y uso de la solución propuesta, así como información de especial interés que de incluirse en otros capítulos de la memoria haría menos clara su lectura.

9.1 Anexo A. Manual de instalación

En este manual de instalación se proporcionarán las indicaciones para instalar la solución en un sistema Windows de 64 bits. Sin embargo todos los módulos propuestos son multiplataforma. En las respectivas webs que se informan se podrían descargar versiones de las herramientas para Linux, recomendándose el uso de Red Hat Enterprise Linux 6.5 o CentOS 6.5.

9.1.1 Descargas

Para la instalación de la solución propuesta se deben descargar de internet los diferentes módulos.

1. **SpagoBI 5.1** – El módulo de Auditoría requiere la instalación de la versión 5.1 de SpagoBI. Para ello bajaremos la versión *All in One* que contiene la aplicación ya desplegada sobre un servidor Tomcat 7.0.47

   http://forge.ow2.org/project/showfiles.php?group_id=204&release_id=5612

1.1 **SpagoBI Server 5.1** – La versión 5.1 All in One de SpagoBI Server viene ya preconfigurada para funcionar tal y como se descarga. Solamente necesita asociar el servidor Tomcat a una base de datos, creándola y cambiando los parámetros de configuración de la Base de Datos.

Ilustración 58: Descarga de SpagoBI Server 5.1

1.2 **Scripts de creación de BD** – Para utilizar la base de datos MySQL se deben descargar unos scripts de creación de base de datos del repositorio de SpagoBI.

Ilustración 59: Decarga de scripts de creación de BD SpagoBI Server 5.1

1.3 **SpagoBI Studio –** Para la generación de reportes de datos y la creación de informes KRI se utiliza una versión de eclipse modificada para su uso con SpagoBI que se descarga en su web.

Ilustración 60: Descarga SpagoBI Studio 5.1

2. **Talend –** Para el módulo de carga de datos de la aplicación se instala el software ETL *Talend Open Studio for data integrationn*

http://www.talend.com/download/talend-open-studio?qt-product_tos_download=3#qt-product_tos_download

Además la aplicación requiere la instalación del siguiente software adicional

1. **MySQL 5.6 –**

http://dev.mysql.com/downloads/installer/

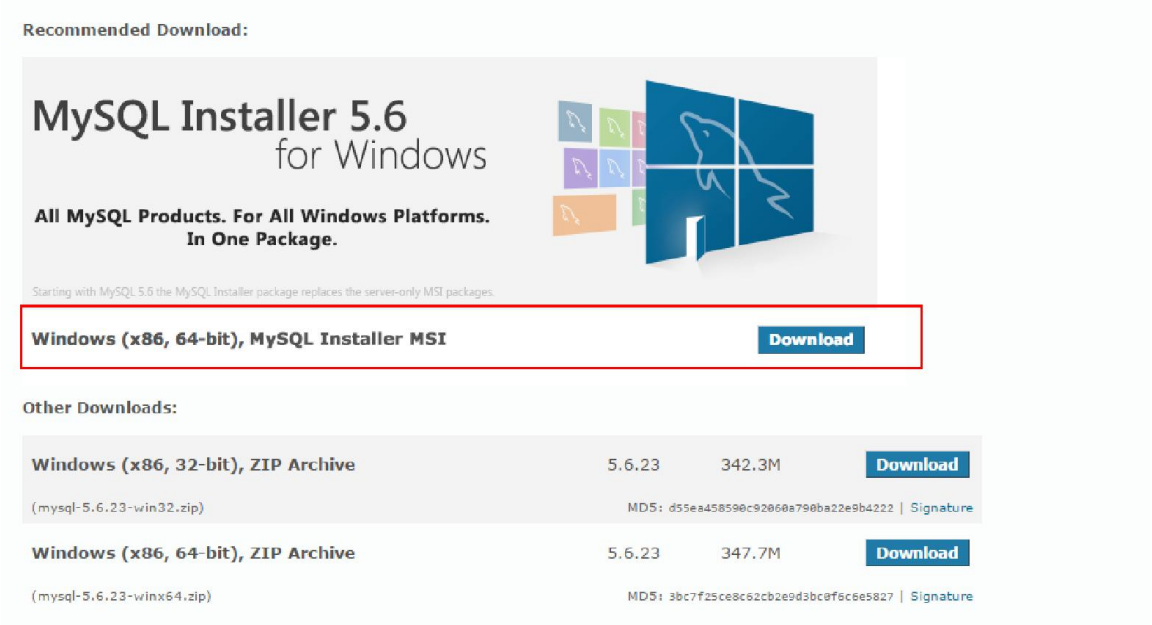


Ilustración 61: Descarga MySQL 5.6

2. **Java JDK 1.7 –** Para el funcionamiento del servidor Tomcat del SpagoBI Server se debe instalar la versión 1.7 del JDK.

http://www.oracle.com/technetwork/es/java/javase/downloads/jdk7-downloads-1880260.html

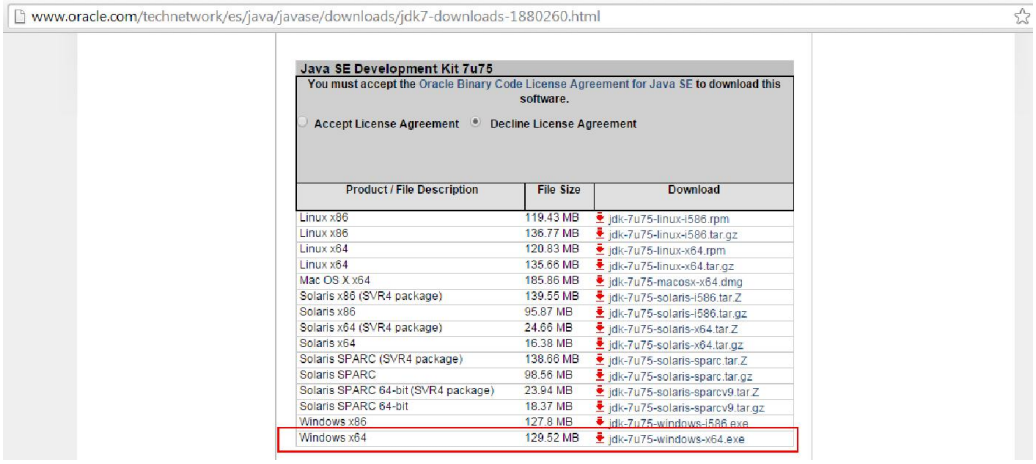


Ilustración 62: Descarga JDK 1.7

3. **MySQL J Connector –** La versión del MySQL J Connector que incluye el SpagoBI server causa problemas con ciertas queries (La versión 5.0.8 no es compatible con las instrucciones SQL `'OPTION SQL_SELECT_LIMIT=DEFAULT'` por lo que se debe actualizar por la versión 5.1.30.

http://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.30

9.1.2 Configuración

1. **MySQL** - Para la configuración del servidor, la primera tarea consiste en montar el servidor de base de datos *MySQL*. La configuración del servidor será la configuración por defecto, no olvidando crear un usuario con permisos de creación, consulta, modificación y borrado de tablas para el uso del SpagoBI.

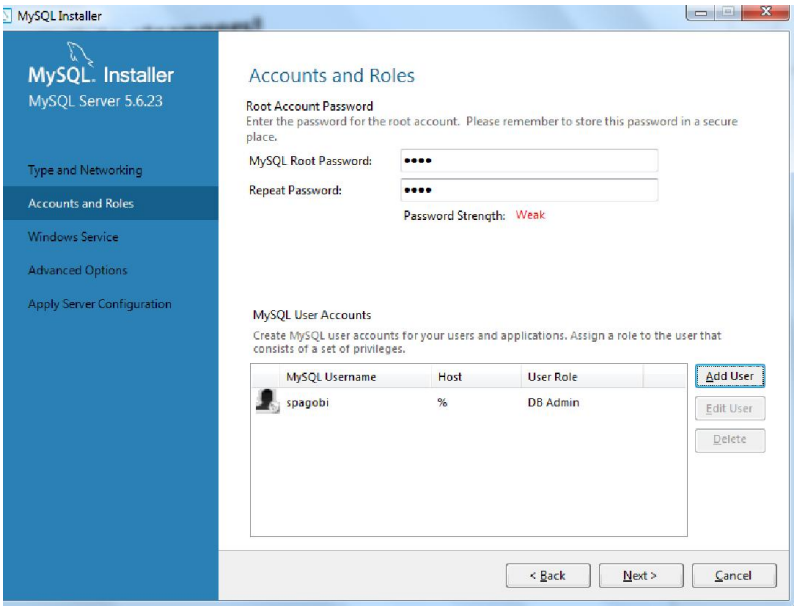


Ilustración 63: Instalación MySQL 5.6

Para la configuración de la base de datos hay dos opciones,

- Primera instalación: En el caso que estemos montando el sistema por primera vez debemos crear la estructura de tablas de la base de datos que utilizará el SpagoBI mediante la ejecución de los scripts SQL descargados anteriormente.

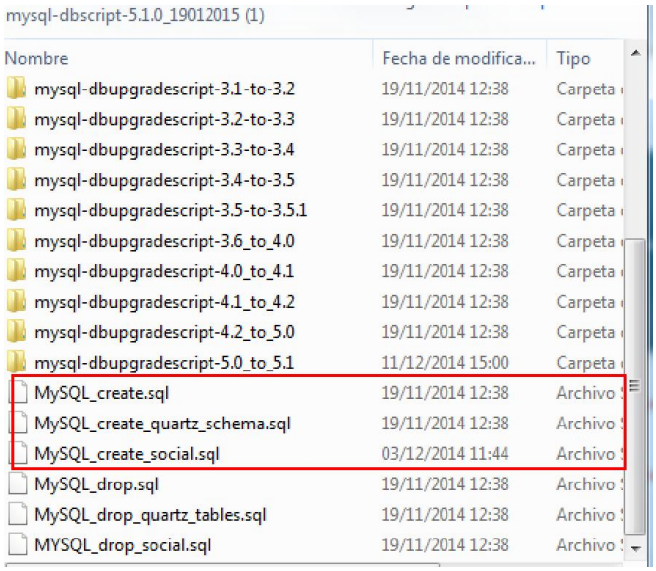


Ilustración 64: Ejecución de scripts creación BD

Para ejecutar los scripts, utilizaremos la interfaz de línea de comandos de MySQL con el comando utilizando un usuario con permisos de root sobre la base de datos. Es importante crear un esquema de base de datos vacío llamado spagobi antes de ejecutar los scripts

```
C:\ > mysql -h localhost -u root -p -D spagobi< MySQL_create.sql
C:\ > mysql -h localhost -u root -p -D spagobi <
MySQL_create_quartz_schema.sql
C:\ > mysql -h localhost -u root -p -D spagobi < MySQL_create_social.sql
```

- Siguientes instalaciones y migraciones: Si lo que pretendemos es montar la solución en posteriores instalaciones, en lugar de ejecutar estos scripts, se restaurará un backup de la base de datos creado mediante la herramienta *mysqldump.* La carga del backup también se realiza ejecutando el script a través de la línea de comandos utilizando un usuario root.

```
C:\ > mysql -h localhost -u root -p -D spagobix< MySQL_spagobi_backup.sql
```

2. **Talend** – Para la configuración del Talend Open Studio, primero ejecutamos el instalador. Para ejecutar Talend debemos utilizar una versión de Java inferior a la 1.8, así que utilizaremos la versión 1.7 que descargamos anteriormente. Para asegurar que se utiliza la JVM que hemos definido lanzaremos el Talend desde un script .bat que definiremos

```
call C:\TalendOS\TOS_DI-win32-x86_64.exe -vm "C:\Program
Files\Java\jdk1.7.0_75\jre\bin"
```

3. **SpagoBI Server** – Para la configuración del servidor, procederemos, primero a descomprimir el archivo *All-In-One-SpagoBI-5.1-21012015.zip* que descargamos. Es importante tener en cuenta que en sistemas Windows la longitud del path hacia un archivo tiene limitación. La carpeta seleccionada para descomprimirlo no debería estar bajo una estructura de carpetas anidadas con nombres largos.

   A continuación se deben modificar los ficheros del *Tomcat* que regulan la conexión a la base de datos a través de *Hibernate* para que el servidor se conecte a nuestra base de datos *MySQL*

   Configuración de la conexión con la Base de datos

   La primera tarea es sustituir el fichero jar del *MySQL J Connector* que viene por defecto por el que nos descargamos para evitar problemas de compatibilidad con MySQL 5.6

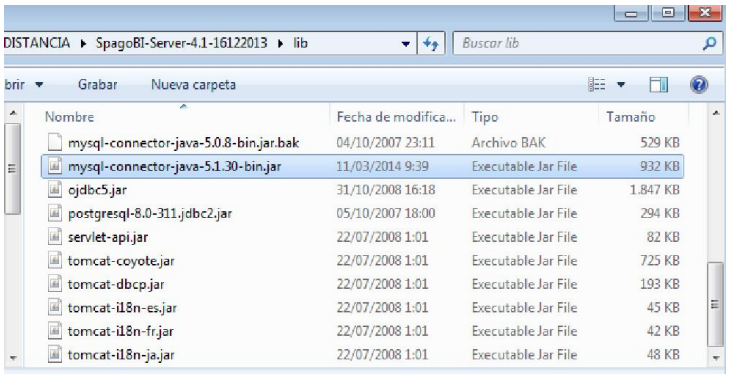


Ilustración 65: MySQL ConnectorJ

A continuación se debe modificar el fichero /conf/server.xml para incorporar el *Datasource* de los datos internos de SPAGOBI. Sustituir el datasource jdbc/spagobi que viene por defecto

```
<Resource type="javax.sql.DataSource" name="jdbc/spagobi"
auth="Container" maxWait="-1" maxIdle="10" maxActive="20"
    password="root" username="root"

url="jdbc:mysql://localhost:3306/spagobi?useOldAliasMetadata
```

```
Behavior=true&amp;autoReconnect=true"
driverClassName="com.mysql.jdbc.Driver"/>
```

Se debe configurar el dialect SQL de Hibernate para que utilice el dialecto específico de MySQL. Esta modificación se debe realizer en los \webapps\SpagoBI\WEB-INF\classes\hibernate.cfg.xml y jbpm.hibernate.cfg.xml

```
<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect
</property>
```

También se debe comentar el dialecto HSQL.

```
<!--    <property
name="hibernate.dialect">org.hibernate.dialect.HSQLDialect<
/property> -->
```

En la misma ruta (\webapps\SpagoBI\WEB-INF\classes\ ) se deben modificar las propiedades del scheduler quartz. Para ello descomentamos la delegate class MySQL y comentamos la HSQL

```
# Mysql/Ingres delegate class
Org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbc
jobstore.StdJDBCDelegate

# Hsqldb delegate class
#org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdb
cjobstore.HSQLDBDelegate
```

Configuración de la variable JAVA_HOME

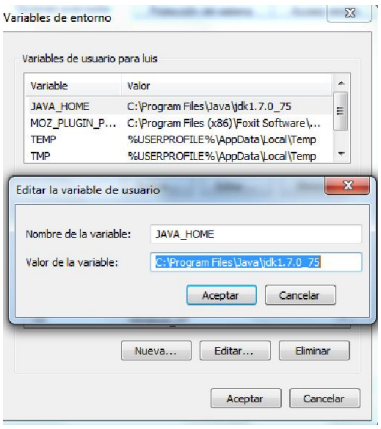


Ilustración 66: Variable de entorno JAVA_HOME

Una vez realizados estos pasos podemos ejecutar el servidor lanzando el script startup.bat que se encuentra en la carpeta \bin.de la instalación de SpagoBI

Ilustración 67: Inicio servidor Tomcat SpagoBI Server

Finalizado el proceso de carga, en la terminal se nos avisará del tiempo que tardó en arrancar

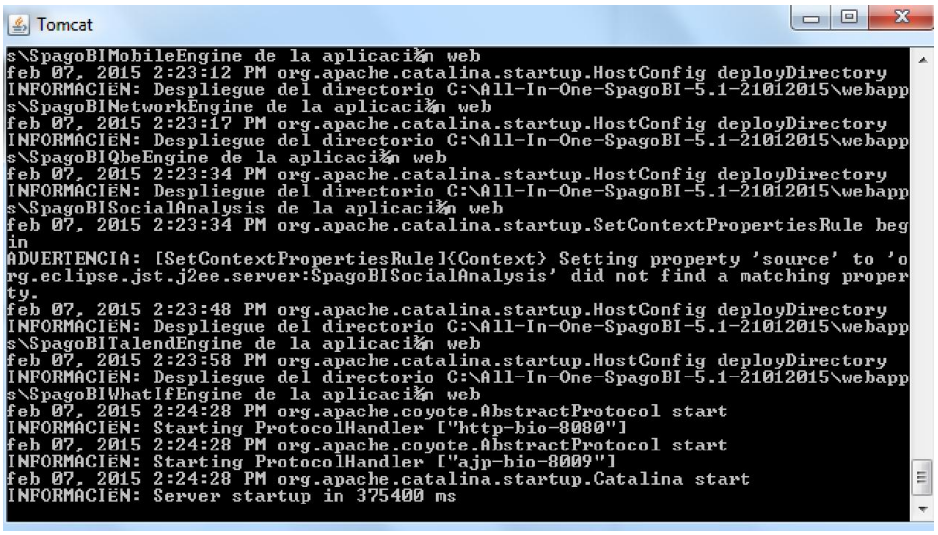


Ilustración 68: Servidor Tomcat iniciado

Tras el arranque accederemos al servidor a través del navegador en la URL http://localhost:8080/SpagoBI

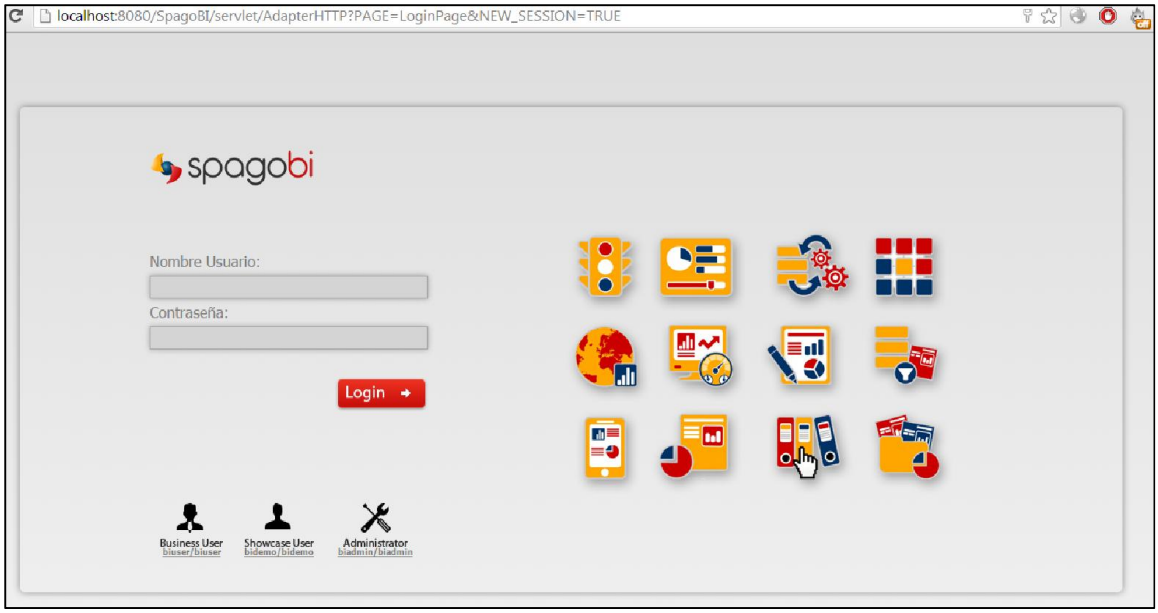


Ilustración 69: Login SpagoBI

Si queremos mantener la personalización de la interfazque hemos realizado, se debería sustituir la carpeta `\webapps\SpagoBI\themes\sbi_default\` y así obtendríamos una vista personalizada tanto del login como de las pantallas de dentro de la aplicación.

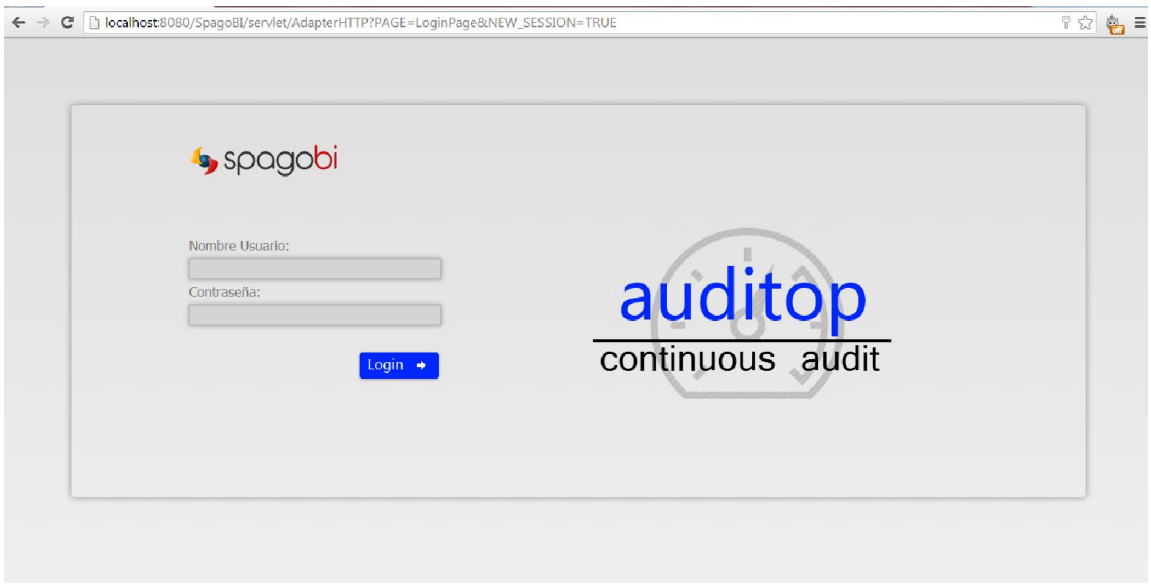


Ilustración 70: Pantalla de Login personalizada

4. **SpagoBI Studio** – Para el desarrollo y modificación de los reportes utilizamos la herramienta de diseño de reportes BI que proporciona SpagoBI. Cuando despleguemos la solución en un nuevo entorno se partirá de una instalación limpia de SpagoBI Studio.

   En el arranque de la aplicación conviene tener en cuenta que en el ordenador que se ejecute debe estar configurada la JVM 1.7, Como la herramienta está basada en una versión modificada de eclipse cuando arranque debemos iniciar la vista SpagoBI.

Ilustración 71: Arranque SpagoBI Studio

El siguiente paso sería crear un proyecto de SpagoBI

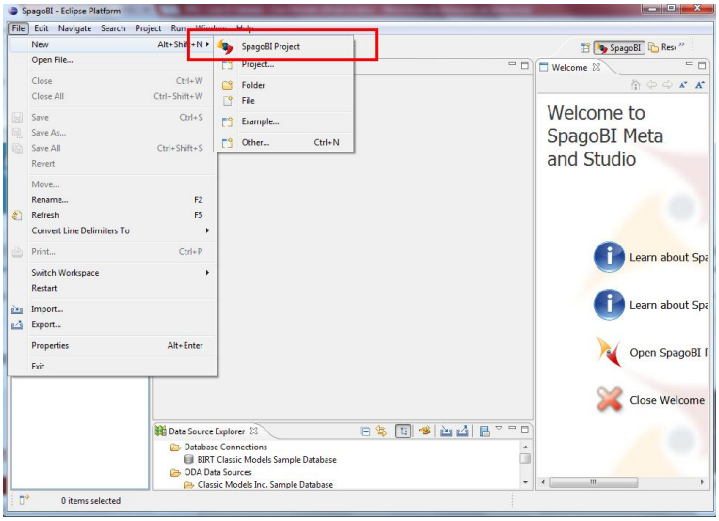


Ilustración 72: Crear proyecto SpagoBI

Para poder realizar las tareas sobre el servidor de nuestra solución, necesitamos configurar la conexión al servidor. Los parámetros de conexión serán: URL, la dirección y puerto por el que accedemos a la interfaz web de SpagoBI. El usuario a utilizar debe tener permisos de administrador.

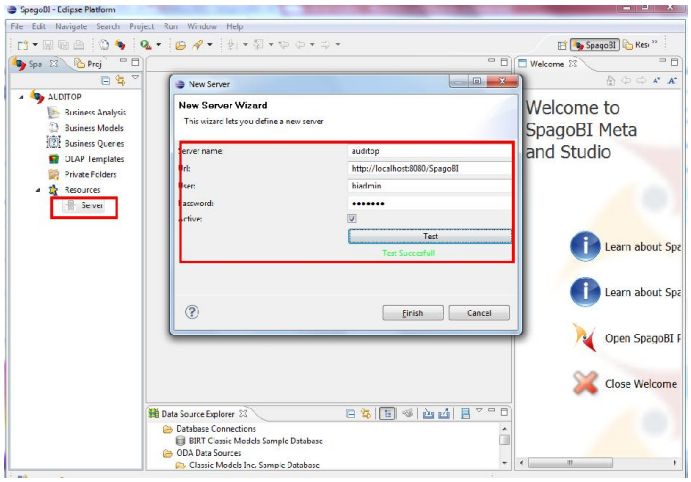


Ilustración 73: Establecer conexión con servidor

El siguiente paso será descargar del servidor los reportes que ya se hayan generado. Esto permitirá que el servidor actue como repositorio centralizado de reportes. Cada vez que se modifique un reporte éste se subirá al servidor utilizando la conexión que hay configurada en el proyecto SpagoBI.

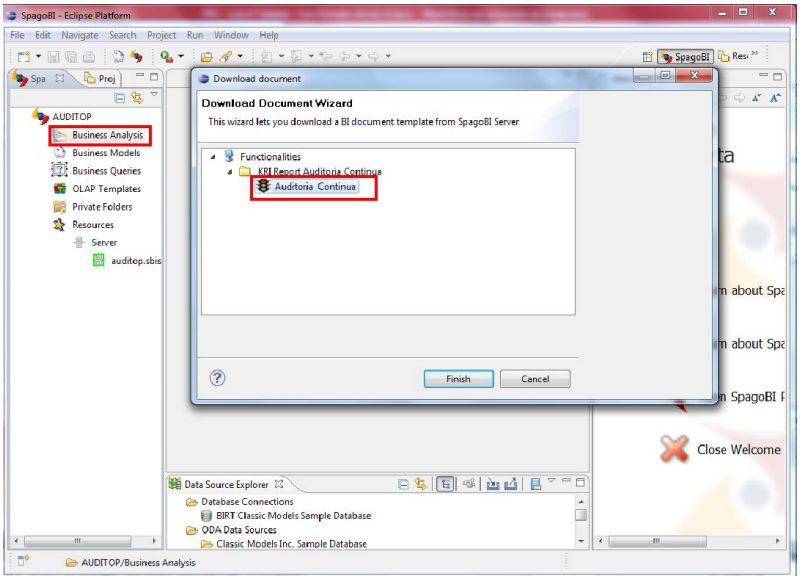


Ilustración 74: Descarga de reportes SpagoBI

Podemos ver como a partir de ahora todos los reportes están disponibles para su modificación.

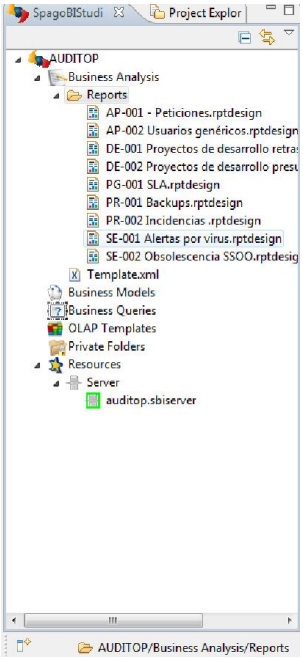


Ilustración 75: Reportes BIRT en SpagoBI Studio